AWS Well-Architected Framework

AWS Well-Architected Framework

AWS Well-Architected Framework

AWS Well-Architected Framework: AWS Well-Architected Framework

Table of Contents

AWS Well-Architected Framework

Publication date: **July 2020** (*Document Revisions* (p. 42))

Abstract

The AWS Well-Architected Framework helps you understand the pros and cons of decisions you make while building systems on AWS. By using the Framework you will learn architectural best practices for designing and operating reliable, secure, efficient, and cost-effective systems in the cloud.

Introduction

The AWS Well-Architected Framework helps you understand the pros and cons of decisions you make while building systems on AWS. By using the Framework you will learn architectural best practices for designing and operating reliable, secure, efficient, and cost-effective systems in the cloud. It provides a way for you to consistently measure your architectures against best practices and identify areas for improvement. The process for reviewing an architecture is a constructive conversation about architectural decisions, and is not an audit mechanism. We believe that having well-architected systems greatly increases the likelihood of business success.

AWS Solutions Architects have years of experience architecting solutions across a wide variety of business verticals and use cases. We have helped design and review thousands of customers' architectures on AWS. From this experience, we have identified best practices and core strategies for architecting systems in the cloud.

The AWS Well-Architected Framework documents a set of foundational questions that allow you to understand if a specific architecture aligns well with cloud best practices. The framework provides a consistent approach to evaluating systems against the qualities you expect from modern cloud-based systems, and the remediation that would be required to achieve those qualities. As AWS continues to evolve, and we continue to learn more from working with our customers, we will continue to refine the definition of well-architected.

This framework is intended for those in technology roles, such as chief technology officers (CTOs), architects, developers, and operations team members. It describes AWS best practices and strategies to use when designing and operating a cloud workload, and provides links to further implementation details and architectural patterns. For more information, see the AWS Well-Architected homepage.

AWS also provides a service for reviewing your workloads at no charge. The AWS Well-Architected Tool (AWS WA Tool) is a service in the cloud that provides a consistent process for you to review and measure your architecture using the AWS Well-Architected Framework. The AWS WA Tool provides recommendations for making your workloads more reliable, secure, efficient, and cost-effective.

To help you apply best practices, we have created AWS Well-Architected Labs, which provides you with a repository of code and documentation to give you hands-on experience implementing best practices. We also have teamed up with select AWS Partner Network (APN) Partners, who are members of the AWS Well-Architected Partner program. These APN Partners have deep AWS knowledge, and can help you review and improve your workloads.

Definitions

Every day, experts at AWS assist customers in architecting systems to take advantage of best practices in the cloud. We work with you on making architectural trade-offs as your designs evolve. As you deploy these systems into live environments, we learn how well these systems perform and the consequences of those trade-offs.

Based on what we have learned, we have created the AWS Well-Architected Framework, which provides a consistent set of best practices for customers and partners to evaluate architectures, and provides a set of questions you can use to evaluate how well an architecture is aligned to AWS best practices.

The AWS Well-Architected Framework is based on five pillars — operational excellence, security, reliability, performance efficiency, and cost optimization.

Table 1. The pillars of the AWS Well-Architected Framework

Name	Description
Operational Excellence	The ability to support development and run workloads effectively, gain insight into their operations, and to continuously improve supporting processes and procedures to deliver business value.
Security	The security pillar describes how to take advantage of cloud technologies to protect data, systems, and assets in a way that can improve your security posture.
Reliability	The reliability pillar encompasses the ability of a workload to perform its intended function correctly and consistently when it's expected to. This includes the ability to operate and test the workload through its total lifecycle. This paper provides in-depth, best practice guidance for implementing reliable workloads on AWS.
Performance Efficiency	The ability to use computing resources efficiently to meet system requirements, and to maintain that efficiency as demand changes and technologies evolve.
Cost Optimization	The ability to run systems to deliver business value at the lowest price point.

In the AWS Well-Architected Framework, we use these terms:

- A **component** is the code, configuration, and AWS Resources that together deliver against a requirement. A component is often the unit of technical ownership, and is decoupled from other components.
- The term **workload** is used to identify a set of components that together deliver business value. A workload is usually the level of detail that business and technology leaders communicate about.
- We think about **architecture** as being how components work together in a workload. How components communicate and interact is often the focus of architecture diagrams.
- **Milestones** mark key changes in your architecture as it evolves throughout the product lifecycle (design, implementation, testing, go live, and in production).
- Within an organization the **technology portfolio** is the collection of workloads that are required for the business to operate.

When architecting workloads, you make trade-offs between pillars based on your business context. These business decisions can drive your engineering priorities. You might optimize to reduce cost at the expense of reliability in development environments, or, for mission-critical solutions, you might optimize reliability with increased costs. In ecommerce solutions, performance can affect revenue and customer propensity to buy. Security and operational excellence are generally not traded-off against the other pillars.

On Architecture

In on-premises environments, customers often have a central team for technology architecture that acts as an overlay to other product or feature teams to ensure they are following best practice. Technology architecture teams typically include a set of roles such as: Technical Architect (infrastructure), Solutions Architect (software), Data Architect, Networking Architect, and Security Architect. Often these teams use TOGAF or the Zachman Framework as part of an enterprise architecture capability.

At AWS, we prefer to distribute capabilities into teams rather than having a centralized team with that capability. There are risks when you choose to distribute decision making authority, for example, ensuring that teams are meeting internal standards. We mitigate these risks in two ways. First, we have *practices* (ways of doing things, process, standards, and accepted norms) that focus on enabling each team to have that capability, and we put in place experts who ensure that teams raise the bar on the standards they need to meet. Second, we implement *mechanisms* that carry out automated checks to ensure standards are being met.

> "Good intentions never work, you need good mechanisms to make anything happen" — Jeff Bezos.

This means replacing humans best efforts with mechanisms (often automated) that check for compliance with rules or process. This distributed approach is supported by the Amazon leadership principles, and establishes a culture across all roles that *works back* from the customer. Working backward is a fundamental part of our innovation process. We start with the customer and what they want, and let that define and guide our efforts. Customer-obsessed teams build products in response to a customer need.

For architecture, this means that we expect every team to have the capability to create architectures and to follow best practices. To help new teams gain these capabilities or existing teams to raise their bar, we enable access to a virtual community of principal engineers who can review their designs and help them understand what AWS best practices are. The principal engineering community works to make best practices visible and accessible. One way they do this, for example, is through lunchtime talks that focus on applying best practices to real examples. These talks are recorded and can be used as part of onboarding materials for new team members.

AWS best practices emerge from our experience running thousands of systems at internet scale. We prefer to use data to define best practice, but we also use subject matter experts, like principal engineers, to set them. As principal engineers see new best practices emerge, they work as a community to ensure that teams follow them. In time, these best practices are formalized into our internal review processes, as well as into mechanisms that enforce compliance. The Well-Architected Framework is the customer-facing implementation of our internal review process, where we have codified our principal engineering thinking across field roles, like Solutions Architecture and internal engineering teams. The Well-Architected Framework is a scalable mechanism that lets you take advantage of these learnings.

By following the approach of a principal engineering community with distributed ownership of architecture, we believe that a Well-Architected enterprise architecture can emerge that is driven by customer need. Technology leaders (such as a CTOs or development managers), carrying out Well-Architected reviews across all your workloads will allow you to better understand the risks in your technology portfolio. Using this approach, you can identify themes across teams that your organization could address by mechanisms, training, or lunchtime talks where your principal engineers can share their thinking on specific areas with multiple teams.

General Design Principles

The Well-Architected Framework identifies a set of general design principles to facilitate good design in the cloud:

- **Stop guessing your capacity needs**: If you make a poor capacity decision when deploying a workload, you might end up sitting on expensive idle resources or dealing with the performance implications of limited capacity. With cloud computing, these problems can go away. You can use as much or as little capacity as you need, and scale up and down automatically.
- **Test systems at production scale**: In the cloud, you can create a production-scale test environment on demand, complete your testing, and then decommission the resources. Because you only pay for the test environment when it's running, you can simulate your live environment for a fraction of the cost of testing on premises.
- **Automate to make architectural experimentation easier**: Automation allows you to create and replicate your workloads at low cost and avoid the expense of manual effort. You can track changes to your automation, audit the impact, and revert to previous parameters when necessary.
- **Allow for evolutionary architectures**: In a traditional environment, architectural decisions are often implemented as static, onetime events, with a few major versions of a system during its lifetime. As a business and its context continue to evolve, these initial decisions might hinder the system's ability to deliver changing business requirements. In the cloud, the capability to automate and test on demand lowers the risk of impact from design changes. This allows systems to evolve over time so that businesses can take advantage of innovations as a standard practice.
- **Drive architectures using data**: In the cloud, you can collect data on how your architectural choices affect the behavior of your workload. This lets you make factbased decisions on how to improve your workload. Your cloud infrastructure is code, so you can use that data to inform your architecture choices and improvements over time.
- **Improve through game days**: Test how your architecture and processes perform by regularly scheduling game days to simulate events in production. This will help you understand where improvements can be made and can help develop organizational experience in dealing with events.

The Five Pillars of the Framework

Creating a software system is a lot like constructing a building. If the foundation is not solid, structural problems can undermine the integrity and function of the building. When architecting technology solutions, if you neglect the five pillars of operational excellence, security, reliability, performance efficiency, and cost optimization, it can become challenging to build a system that delivers on your expectations and requirements. Incorporating these pillars into your architecture will help you produce stable and efficient systems. This will allow you to focus on the other aspects of design, such as functional requirements.

Topics

Operational Excellence

The Operational Excellence pillar includes the ability to support development and run workloads effectively, gain insight into their operations, and to continuously improve supporting processes and procedures to deliver business value.

The operational excellence pillar provides an overview of design principles, best practices, and questions. You can find prescriptive guidance on implementation in the Operational Excellence Pillar whitepaper.

Topics

Design Principles

There are five design principles for operational excellence in the cloud:

- **Perform operations as code**: In the cloud, you can apply the same engineering discipline that you use for application code to your entire environment. You can define your entire workload (applications, infrastructure) as code and update it with code. You can implement your operations procedures as code and automate their execution by triggering them in response to events. By performing operations as code, you limit human error and enable consistent responses to events.
- **Make frequent, small, reversible changes**: Design workloads to allow components to be updated regularly. Make changes in small increments that can be reversed if they fail (without affecting customers when possible).
- **Refine operations procedures frequently**: As you use operations procedures, look for opportunities to improve them. As you evolve your workload, evolve your procedures appropriately. Set up regular

game days to review and validate that all procedures are effective and that teams are familiar with them.

- **Anticipate failure**: Perform “pre-mortem” exercises to identify potential sources of failure so that they can be removed or mitigated. Test your failure scenarios and validate your understanding of their impact. Test your response procedures to ensure that they are effective, and that teams are familiar with their execution. Set up regular game days to test workloads and team responses to simulated events.
- **Learn from all operational failures**: Drive improvement through lessons learned from all operational events and failures. Share what is learned across teams and through the entire organization.

Definition

There are four best practice areas for operational excellence in the cloud:

- **Organization**
- **Prepare**
- **Operate**
- **Evolve**

Your organization’s leadership defines business objectives. Your organization must understand requirements and priorities and use these to organize and conduct work to support the achievement of business outcomes. Your workload must emit the information necessary to support it. Implementing services to enable integration, deployment, and delivery of your workload will enable an increased flow of beneficial changes into production by automating repetitive processes.

There may be risks inherent in the operation of your workload. You must understand those risks and make an informed decision to enter production. Your teams must be able to support your workload. Business and operational metrics derived from desired business outcomes will enable you to understand the health of your workload, your operations activities, and respond to incidents. Your priorities will change as your business needs and business environment changes. Use these as a feedback loop to continually drive improvement for your organization and the operation of your workload.

Best Practices

Topics

Organization

Your teams need to have a shared understanding of your entire workload, their role in it, and shared business goals to set the priorities that will enable business success. Well-defined priorities will maximize the benefits of your efforts. Evaluate internal and external customer needs involving key stakeholders, including business, development, and operations teams, to determine where to focus efforts. Evaluating customer needs will ensure that you have a thorough understanding of the support that is required to achieve business outcomes. Ensure that you are aware of guidelines or obligations defined by your organizational governance and external factors, such as regulatory compliance requirements and industry standards, that may mandate or emphasize specific focus. Validate that you have mechanisms to identify changes to internal governance and external compliance requirements. If no requirements

are identified, ensure that you have applied due diligence to this determination. Review your priorities regularly so that they can be updated as needs change.

Evaluate threats to the business (for example, business risk and liabilities, and information security threats) and maintain this information in a risk registry. Evaluate the impact of risks, and tradeoffs between competing interests or alternative approaches. For example, accelerating speed to market for new features may be emphasized over cost optimization, or you may choose a relational database for non-relational data to simplify the effort to migrate a system without refactoring. Manage benefits and risks to make informed decisions when determining where to focus efforts. Some risks or choices may be acceptable for a time, it may be possible to mitigate associated risks, or it may become unacceptable to allow a risk to remain, in which case you will take action to address the risk.

Your teams must understand their part in achieving business outcomes. Teams need to understand their roles in the success of other teams, the role of other teams in their success, and have shared goals. Understanding responsibility, ownership, how decisions are made, and who has authority to make decisions will help focus efforts and maximize the benefits from your teams. The needs of a team will be shaped by the customer they support, their organization, the makeup of the team, and the characteristics of their workload. It's unreasonable to expect a single operating model to be able to support all teams and their workloads in your organization.

Ensure that there are identified owners for each application, workload, platform, and infrastructure component, and that each process and procedure has an identified owner responsible for its definition, and owners responsible for their performance.

Having understanding of the business value of each component, process, and procedure, of why those resources are in place or activities are performed, and why that ownership exists will inform the actions of your team members. Clearly define the responsibilities of team members so that they may act appropriately and have mechanisms to identify responsibility and ownership. Have mechanisms to request additions, changes, and exceptions so that you do not constrain innovation. Define agreements between teams describing how they work together to support each other and your business outcomes.

Provide support for your team members so that they can be more effective in taking action and supporting your business outcomes. Engaged senior leadership should set expectations and measure success. They should be the sponsor, advocate, and driver for the adoption of best practices and evolution of the organization. Empower team members to take action when outcomes are at risk to minimize impact and encourage them to escalate to decision makers and stakeholders when they believe there is a risk so that it can be addressed and incidents avoided. Provide timely, clear, and actionable communications of known risks and planned events so that team members can take timely and appropriate action.

Encourage experimentation to accelerate learning and keep team members interested and engaged. Teams must grow their skill sets to adopt new technologies, and to support changes in demand and responsibilities. Support and encourage this by providing dedicated structured time for learning. Ensure your team members have the resources, both tools and team members, to be successful and scale to support your business outcomes. Leverage cross-organizational diversity to seek multiple unique perspectives. Use this perspective to increase innovation, challenge your assumptions, and reduce the risk of confirmation bias. Grow inclusion, diversity, and accessibility within your teams to gain beneficial perspectives.

If there are external regulatory or compliance requirements that apply to your organization, you should use the resources provided by AWS Cloud Compliance to help educate your teams so that they can determine the impact on your priorities. The Well-Architected Framework emphasizes learning, measuring, and improving. It provides a consistent approach for you to evaluate architectures, and implement designs that will scale over time. AWS provides the AWS Well-Architected Tool to help you review your approach prior to development, the state of your workloads prior to production, and the state of your workloads in production. You can compare workloads to the latest AWS architectural best practices, monitor their overall status, and gain insight into potential risks. AWS Trusted Advisor is a tool that provides access to a core set of checks that recommend optimizations that may help shape your

priorities. Business and Enterprise Support customers receive access to additional checks focusing on security, reliability, performance, and cost-optimization that can further help shape their priorities.

AWS can help you educate your teams about AWS and its services to increase their understanding of how their choices can have an impact on your workload. You should use the resources provided by AWS Support (AWS Knowledge Center, AWS Discussion Forums, and AWS Support Center) and AWS Documentation to educate your teams. Reach out to AWS Support through AWS Support Center for help with your AWS questions. AWS also shares best practices and patterns that we have learned through the operation of AWS in The Amazon Builders' Library. A wide variety of other useful information is available through the AWS Blog and The Official AWS Podcast. AWS Training and Certification provides some free training through self-paced digital courses on AWS fundamentals. You can also register for instructor-led training to further support the development of your teams' AWS skills.

You should use tools or services that enable you to centrally govern your environments across accounts, such as AWS Organizations, to help manage your operating models. Services like AWS Control Tower expand this management capability by enabling you to define blueprints (supporting your operating models) for the setup of accounts, apply ongoing governance using AWS Organizations, and automate provisioning of new accounts. Managed Services providers such as AWS Managed Services, AWS Managed Services Partners, or Managed Services Providers in the AWS Partner Network, provide expertise implementing cloud environments, and support your security and compliance requirements and business goals. Adding Managed Services to your operating model can save you time and resources, and lets you keep your internal teams lean and focused on strategic outcomes that will differentiate your business, rather than developing new skills and capabilities.

The following questions focus on these considerations for operational excellence. (For a list of operational excellence questions and best practices, see the Appendix (p. 44).).

OPS 1: How do you determine what your priorities are?
Everyone needs to understand their part in enabling business success. Have shared goals in order to set priorities for resources. This will maximize the benefits of your efforts.

OPS 2: How do you structure your organization to support your business outcomes?
Your teams must understand their part in achieving business outcomes. Teams need to understand their roles in the success of other teams, the role of other teams in their success, and have shared goals. Understanding responsibility, ownership, how decisions are made, and who has authority to make decisions will help focus efforts and maximize the benefits from your teams.

OPS 3: How does your organizational culture support your business outcomes?
Provide support for your team members so that they can be more effective in taking action and supporting your business outcome.

You might find that you want to emphasize a small subset of your priorities at some point in time. Use a balanced approach over the long term to ensure the development of needed capabilities and management of risk. Review your priorities regularly and update them as needs change. When responsibility and ownership are undefined or unknown, you are at risk of both not performing necessary action in a timely fashion and of redundant and potentially conflicting efforts emerging to address those needs. Organizational culture has a direct impact on team member job satisfaction and retention. Enable the engagement and capabilities of your team members to enable the success of your business. Experimentation is required for innovation to happen and turn ideas into outcomes. Recognize that an undesired result is a successful experiment that has identified a path that will not lead to success.

Prepare

To prepare for operational excellence, you have to understand your workloads and their expected behaviors. You will then be able to design them to provide insight to their status and build the procedures to support them.

Design your workload so that it provides the information necessary for you to understand its internal state (for example, metrics, logs, events, and traces) across all components in support of observability and investigating issues. Iterate to develop the telemetry necessary to monitor the health of your workload, identify when outcomes are at risk, and enable effective responses. When instrumenting your workload, capture a broad set of information to enable situational awareness (for example, changes in state, user activity, privilege access, utilization counters), knowing that you can use filters to select the most useful information over time.

Adopt approaches that improve the flow of changes into production and that enable refactoring, fast feedback on quality, and bug fixing. These accelerate beneficial changes entering production, limit issues deployed, and enable rapid identification and remediation of issues introduced through deployment activities or discovered in your environments.

Adopt approaches that provide fast feedback on quality and enable rapid recovery from changes that do not have desired outcomes. Using these practices mitigates the impact of issues introduced through the deployment of changes. Plan for unsuccessful changes so that you are able to respond faster if necessary and test and validate the changes you make. Be aware of planned activities in your environments so that you can manage the risk of changes impacting planned activities. Emphasize frequent, small, reversible changes to limit the scope of change. This results in easier troubleshooting and faster remediation with the option to roll back a change. It also means you are able to get the benefit of valuable changes more frequently.

Evaluate the operational readiness of your workload, processes, procedures, and personnel to understand the operational risks related to your workload. You should use a consistent process (including manual or automated checklists) to know when you are ready to go live with your workload or a change. This will also enable you to find any areas that you need to make plans to address. Have runbooks that document your routine activities and playbooks that guide your processes for issue resolution. Understand the benefits and risks to make informed decisions to allow changes to enter production.

AWS enables you to view your entire workload (applications, infrastructure, policy, governance, and operations) as code. This means you can apply the same engineering discipline that you use for application code to every element of your stack and share these across teams or organizations to magnify the benefits of development efforts. Use operations as code in the cloud and the ability to safely experiment to develop your workload, your operations procedures, and practice failure. Using AWS CloudFormation enables you to have consistent, templated, sandbox development, test, and production environments with increasing levels of operations control.

The following questions focus on these considerations for operational excellence.

OPS 4: How do you design your workload so that you can understand its state?
Design your workload so that it provides the information necessary across all components (for example, metrics, logs, and traces) for you to understand its internal state. This enables you to provide effective responses when appropriate.

OPS 5: How do you reduce defects, ease remediation, and improve flow into production?
Adopt approaches that improve flow of changes into production, that enable refactoring, fast feedback on quality, and bug fixing. These accelerate beneficial changes entering production, limit issues

OPS 5: How do you reduce defects, ease remediation, and improve flow into production?
deployed, and enable rapid identification and remediation of issues introduced through deployment activities.

OPS 6: How do you mitigate deployment risks?
Adopt approaches that provide fast feedback on quality and enable rapid recovery from changes that do not have desired outcomes. Using these practices mitigates the impact of issues introduced through the deployment of changes.

OPS 7: How do you know that you are ready to support a workload?
Evaluate the operational readiness of your workload, processes and procedures, and personnel to understand the operational risks related to your workload.

Invest in implementing operations activities as code to maximize the productivity of operations personnel, minimize error rates, and enable automated responses. Use “pre-mortems” to anticipate failure and create procedures where appropriate. Apply metadata using Resource Tags and AWS Resource Groups following a consistent tagging strategy to enable identification of your resources. Tag your resources for organization, cost accounting, access controls, and targeting the execution of automated operations activities. Adopt deployment practices that take advantage of the elasticity of the cloud to facilitate development activities, and pre-deployment of systems for faster implementations. When you make changes to the checklists you use to evaluate your workloads, plan what you will do with live systems that no longer comply.

Operate

Successful operation of a workload is measured by the achievement of business and customer outcomes. Define expected outcomes, determine how success will be measured, and identify metrics that will be used in those calculations to determine if your workload and operations are successful. Operational health includes both the health of the workload and the health and success of the operations activities performed in support of the workload (for example, deployment and incident response). Establish metrics baselines for improvement, investigation, and intervention, collect and analyze your metrics, and then validate your understanding of operations success and how it changes over time. Use collected metrics to determine if you are satisfying customer and business needs, and identify areas for improvement.

Efficient and effective management of operational events is required to achieve operational excellence. This applies to both planned and unplanned operational events. Use established runbooks for well-understood events, and use playbooks to aid in investigation and resolution of issues. Prioritize responses to events based on their business and customer impact. Ensure that if an alert is raised in response to an event, there is an associated process to be executed, with a specifically identified owner. Define in advance the personnel required to resolve an event and include escalation triggers to engage additional personnel, as it becomes necessary, based on urgency and impact. Identify and engage individuals with the authority to make a decision on courses of action where there will be a business impact from an event response not previously addressed.

Communicate the operational status of workloads through dashboards and notifications that are tailored to the target audience (for example, customer, business, developers, operations) so that they may take appropriate action, so that their expectations are managed, and so that they are informed when normal operations resume.

In AWS, you can generate dashboard views of your metrics collected from workloads and natively from AWS. You can leverage CloudWatch or third-party applications to aggregate and present business, workload, and operations level views of operations activities. AWS provides workload insights through logging capabilities including AWS X-Ray, CloudWatch, CloudTrail, and VPC Flow Logs enabling the identification of workload issues in support of root cause analysis and remediation.

The following questions focus on these considerations for operational excellence.

OPS 8: How do you understand the health of your workload?
Define, capture, and analyze workload metrics to gain visibility to workload events so that you can take appropriate action.

OPS 9: How do you understand the health of your operations?
Define, capture, and analyze operations metrics to gain visibility to operations events so that you can take appropriate action.

OPS 10: How do you manage workload and operations events?
Prepare and validate procedures for responding to events to minimize their disruption to your workload.

All of the metrics you collect should be aligned to a business need and the outcomes they support. Develop scripted responses to well-understood events and automate their performance in response to recognizing the event.

Evolve

You must learn, share, and continuously improve to sustain operational excellence. Dedicate work cycles to making continuous incremental improvements. Perform post-incident analysis of all customer impacting events. Identify the contributing factors and preventative action to limit or prevent recurrence. Communicate contributing factors with affected communities as appropriate. Regularly evaluate and prioritize opportunities for improvement (for example, feature requests, issue remediation, and compliance requirements), including both the workload and operations procedures.

Include feedback loops within your procedures to rapidly identify areas for improvement and capture learnings from the execution of operations.

Share lessons learned across teams to share the benefits of those lessons. Analyze trends within lessons learned and perform cross-team retrospective analysis of operations metrics to identify opportunities and methods for improvement. Implement changes intended to bring about improvement and evaluate the results to determine success.

On AWS, you can export your log data to Amazon S3 or send logs directly to Amazon S3 for long-term storage. Using AWS Glue, you can discover and prepare your log data in Amazon S3 for analytics, and store associated metadata in the AWS Glue Data Catalog. Amazon Athena, through its native integration with AWS Glue, can then be used to analyze your log data, querying it using standard SQL. Using a business intelligence tool like Amazon QuickSight, you can visualize, explore, and analyze your data. Discovering trends and events of interest that may drive improvement.

The following question focuses on these considerations for operational excellence.

OPS 11: How do you evolve operations?
Dedicate time and resources for continuous incremental improvement to evolve the effectiveness and efficiency of your operations.

Successful evolution of operations is founded in: frequent small improvements; providing safe environments and time to experiment, develop, and test improvements; and environments in which learning from failures is encouraged. Operations support for sandbox, development, test, and production environments, with increasing level of operational controls, facilitates development and increases the predictability of successful results from changes deployed into production.

Resources

Refer to the following resources to learn more about our best practices for Operational Excellence.

Documentation

- DevOps and AWS

Whitepaper

- Operational Excellence Pillar

Video

- DevOps at Amazon

Security

The Security pillar encompasses the ability to protect data, systems, and assets to take advantage of cloud technologies to improve your security.

The security pillar provides an overview of design principles, best practices, and questions. You can find prescriptive guidance on implementation in the Security Pillar whitepaper.

Topics

Design Principles

There are seven design principles for security in the cloud:

- **Implement a strong identity foundation**: Implement the principle of least privilege and enforce separation of duties with appropriate authorization for each interaction with your AWS resources. Centralize identity management, and aim to eliminate reliance on long-term static credentials.

- **Enable traceability**: Monitor, alert, and audit actions and changes to your environment in real time. Integrate log and metric collection with systems to automatically investigate and take action.
- **Apply security at all layers**: Apply a defense in depth approach with multiple security controls. Apply to all layers (for example, edge of network, VPC, load balancing, every instance and compute service, operating system, application, and code).
- **Automate security best practices**: Automated software-based security mechanisms improve your ability to securely scale more rapidly and cost-effectively. Create secure architectures, including the implementation of controls that are defined and managed as code in version-controlled templates.
- **Protect data in transit and at rest**: Classify your data into sensitivity levels and use mechanisms, such as encryption, tokenization, and access control where appropriate.
- **Keep people away from data**: Use mechanisms and tools to reduce or eliminate the need for direct access or manual processing of data. This reduces the risk of mishandling or modification and human error when handling sensitive data.
- **Prepare for security events**: Prepare for an incident by having incident management and investigation policy and processes that align to your organizational requirements. Run incident response simulations and use tools with automation to increase your speed for detection, investigation, and recovery.

Definition

There are six best practice areas for security in the cloud:

- **Security**
- **Identity and Access Management**
- **Detection**
- **Infrastructure Protection**
- **Data Protection**
- **Incident Response**

Before you architect any workload, you need to put in place practices that influence security. You will want to control who can do what. In addition, you want to be able to identify security incidents, protect your systems and services, and maintain the confidentiality and integrity of data through data protection. You should have a well-defined and practiced process for responding to security incidents. These tools and techniques are important because they support objectives such as preventing financial loss or complying with regulatory obligations.

The AWS Shared Responsibility Model enables organizations that adopt the cloud to achieve their security and compliance goals. Because AWS physically secures the infrastructure that supports our cloud services, as an AWS customer you can focus on using services to accomplish your goals. The AWS Cloud also provides greater access to security data and an automated approach to responding to security events.

Best Practices

Topics

- Incident Response (p. 18)

Security

To operate your workload securely, you must apply overarching best practices to every area of security. Take requirements and processes that you have defined in operational excellence at an organizational and workload level, and apply them to all areas.

Staying up to date with AWS and industry recommendations and threat intelligence helps you evolve your threat model and control objectives. Automating security processes, testing, and validation allow you to scale your security operations.

The following question focuses on these considerations for security. (For a list of security questions and best practices, see the Appendix (p. 51).).

SEC 1: How do you securely operate your workload?
To operate your workload securely, you must apply overarching best practices to every area of security. Take requirements and processes that you have defined in operational excellence at an organizational and workload level, and apply them to all areas. Staying up to date with recommendations from AWS, industry sources, and threat intelligence helps you evolve your threat model and control objectives. Automating security processes, testing, and validation allow you to scale your security operations.

In AWS, segregating different workloads by account, based on their function and compliance or data sensitivity requirements, is a recommended approach.

Identity and Access Management

Identity and access management are key parts of an information security program, ensuring that only authorized and authenticated users and components are able to access your resources, and only in a manner that you intend. For example, you should define principals (that is, accounts, users, roles, and services that can perform actions in your account), build out policies aligned with these principals, and implement strong credential management. These privilege-management elements form the core of authentication and authorization.

In AWS, privilege management is primarily supported by the AWS Identity and Access Management (IAM) service, which allows you to control user and programmatic access to AWS services and resources. You should apply granular policies, which assign permissions to a user, group, role, or resource. You also have the ability to require strong password practices, such as complexity level, avoiding re-use, and enforcing multi-factor authentication (MFA). You can use federation with your existing directory service. For workloads that require systems to have access to AWS, IAM enables secure access through roles, instance profiles, identity federation, and temporary credentials.

The following questions focus on these considerations for security.

SEC 2: How do you manage identities for people and machines?
There are two types of identities you need to manage when approaching operating secure AWS workloads. Understanding the type of identity you need to manage and grant access helps you ensure the right identities have access to the right resources under the right conditions. Human Identities: Your administrators, developers, operators, and end users require an identity to access your AWS environments and applications. These are members of your organization, or external

SEC 2: How do you manage identities for people and machines?
users with whom you collaborate, and who interact with your AWS resources via a web browser, client application, or interactive command-line tools. Machine Identities: Your service applications, operational tools, and workloads require an identity to make requests to AWS services, for example, to read data. These identities include machines running in your AWS environment such as Amazon EC2 instances or AWS Lambda functions. You may also manage machine identities for external parties who need access. Additionally, you may also have machines outside of AWS that need access to your AWS environment.

SEC 3: How do you manage permissions for people and machines?
Manage permissions to control access to people and machine identities that require access to AWS and your workload. Permissions control who can access what, and under what conditions.

Credentials must not be shared between any user or system. User access should be granted using a least-privilege approach with best practices including password requirements and MFA enforced. Programmatic access including API calls to AWS services should be performed using temporary and limited-privilege credentials such as those issued by the AWS Security Token Service.

AWS provides resources that can help you with Identity and access management. To help learn best practices, explore our hands-on labs on managing credentials & authentication, controlling human access, and controlling programmatic access.

Detection

You can use detective controls to identify a potential security threat or incident. They are an essential part of governance frameworks and can be used to support a quality process, a legal or compliance obligation, and for threat identification and response efforts. There are different types of detective controls. For example, conducting an inventory of assets and their detailed attributes promotes more effective decision making (and lifecycle controls) to help establish operational baselines. You can also use internal auditing, an examination of controls related to information systems, to ensure that practices meet policies and requirements and that you have set the correct automated alerting notifications based on defined conditions. These controls are important reactive factors that can help your organization identify and understand the scope of anomalous activity.

In AWS, you can implement detective controls by processing logs, events, and monitoring that allows for auditing, automated analysis, and alarming. CloudTrail logs, AWS API calls, and CloudWatch provide monitoring of metrics with alarming, and AWS Config provides configuration history. Amazon GuardDuty is a managed threat detection service that continuously monitors for malicious or unauthorized behavior to help you protect your AWS accounts and workloads. Service-level logs are also available, for example, you can use Amazon Simple Storage Service (Amazon S3) to log access requests.

The following question focuses on these considerations for security.

SEC 4: How do you detect and investigate security events?
Capture and analyze events from logs and metrics to gain visibility. Take action on security events and potential threats to help secure your workload.

Log management is important to a Well-Architected workload for reasons ranging from security or forensics to regulatory or legal requirements. It is critical that you analyze logs and respond to

them so that you can identify potential security incidents. AWS provides functionality that makes log management easier to implement by giving you the ability to define a data-retention lifecycle or define where data will be preserved, archived, or eventually deleted. This makes predictable and reliable data handling simpler and more cost effective.

Infrastructure Protection

Infrastructure protection encompasses control methodologies, such as defense in depth, necessary to meet best practices and organizational or regulatory obligations. Use of these methodologies is critical for successful, ongoing operations in either the cloud or on-premises.

In AWS, you can implement stateful and stateless packet inspection, either by using AWS-native technologies or by using partner products and services available through the AWS Marketplace. You should use Amazon Virtual Private Cloud (Amazon VPC) to create a private, secured, and scalable environment in which you can define your topology—including gateways, routing tables, and public and private subnets.

The following questions focus on these considerations for security.

SEC 5: How do you protect your network resources?
Any workload that has some form of network connectivity, whether it's the internet or a private network, requires multiple layers of defense to help protect from external and internal network-based threats.

SEC 6: How do you protect your compute resources?
Compute resources in your workload require multiple layers of defense to help protect from external and internal threats. Compute resources include EC2 instances, containers, AWS Lambda functions, database services, IoT devices, and more.

Multiple layers of defense are advisable in any type of environment. In the case of infrastructure protection, many of the concepts and methods are valid across cloud and on-premises models. Enforcing boundary protection, monitoring points of ingress and egress, and comprehensive logging, monitoring, and alerting are all essential to an effective information security plan.

AWS customers are able to tailor, or harden, the configuration of an Amazon Elastic Compute Cloud (Amazon EC2), Amazon Elastic Container Service (Amazon ECS) container, or AWS Elastic Beanstalk instance, and persist this configuration to an immutable Amazon Machine Image (AMI). Then, whether triggered by Auto Scaling or launched manually, all new virtual servers (instances) launched with this AMI receive the hardened configuration.

Data Protection

Before architecting any system, foundational practices that influence security should be in place. For example, data classification provides a way to categorize organizational data based on levels of sensitivity, and encryption protects data by way of rendering it unintelligible to unauthorized access. These tools and techniques are important because they support objectives such as preventing financial loss or complying with regulatory obligations.

In AWS, the following practices facilitate protection of data:

- As an AWS customer you maintain full control over your data.

- AWS makes it easier for you to encrypt your data and manage keys, including regular key rotation, which can be easily automated by AWS or maintained by you.
- Detailed logging that contains important content, such as file access and changes, is available.
- AWS has designed storage systems for exceptional resiliency. For example, Amazon S3 Standard, S3 Standard–IA, S3 One Zone-IA, and Amazon Glacier are all designed to provide 99.999999999% durability of objects over a given year. This durability level corresponds to an average annual expected loss of 0.000000001% of objects.
- Versioning, which can be part of a larger data lifecycle management process, can protect against accidental overwrites, deletes, and similar harm.
- AWS never initiates the movement of data between Regions. Content placed in a Region will remain in that Region unless you explicitly enable a feature or leverage a service that provides that functionality.

The following questions focus on these considerations for security.

SEC 7: How do you classify your data?
Classification provides a way to categorize data, based on criticality and sensitivity in order to help you determine appropriate protection and retention controls.

SEC 8: How do you protect your data at rest?
Protect your data at rest by implementing multiple controls, to reduce the risk of unauthorized access or mishandling.

SEC 9: How do you protect your data in transit?
Protect your data in transit by implementing multiple controls to reduce the risk of unauthorized access or loss.

AWS provides multiple means for encrypting data at rest and in transit. We build features into our services that make it easier to encrypt your data. For example, we have implemented server-side encryption (SSE) for Amazon S3 to make it easier for you to store your data in an encrypted form. You can also arrange for the entire HTTPS encryption and decryption process (generally known as SSL termination) to be handled by Elastic Load Balancing (ELB).

Incident Response

Even with extremely mature preventive and detective controls, your organization should still put processes in place to respond to and mitigate the potential impact of security incidents. The architecture of your workload strongly affects the ability of your teams to operate effectively during an incident, to isolate or contain systems, and to restore operations to a known good state. Putting in place the tools and access ahead of a security incident, then routinely practicing incident response through game days, will help you ensure that your architecture can accommodate timely investigation and recovery.

In AWS, the following practices facilitate effective incident response:

- Detailed logging is available that contains important content, such as file access and changes.
- Events can be automatically processed and trigger tools that automate responses through the use of AWS APIs.

- You can pre-provision tooling and a “clean room” using AWS CloudFormation. This allows you to carry out forensics in a safe, isolated environment.

The following question focuses on these considerations for security.

SEC 10: How do you anticipate, respond to, and recover from incidents?
Preparation is critical to timely and effective investigation, response to, and recovery from security incidents to help minimize disruption to your organization.

Ensure that you have a way to quickly grant access for your security team, and automate the isolation of instances as well as the capturing of data and state for forensics.

Resources

Refer to the following resources to learn more about our best practices for Security.

Documentation

- AWS Cloud Security
- AWS Compliance
- AWS Security Blog

Whitepaper

- Security Pillar
- AWS Security Overview
- AWS Security Best Practices
- AWS Risk and Compliance

Video

- AWS Security State of the Union
- Shared Responsibility Overview

Reliability

The Reliability pillar encompasses the ability of a workload to perform its intended function correctly and consistently when it’s expected to. This includes the ability to operate and test the workload through its total lifecycle. This paper provides in-depth, best practice guidance for implementing reliable workloads on AWS.

The reliability pillar provides an overview of design principles, best practices, and questions. You can find prescriptive guidance on implementation in the Reliability Pillar whitepaper.

Topics

Design Principles

There are five design principles for reliability in the cloud:

- **Automatically recover from failure**: By monitoring a workload for key performance indicators (KPIs), you can trigger automation when a threshold is breached. These KPIs should be a measure of business value, not of the technical aspects of the operation of the service. This allows for automatic notification and tracking of failures, and for automated recovery processes that work around or repair the failure. With more sophisticated automation, it's possible to anticipate and remediate failures before they occur.
- **Test recovery procedures**: In an on-premises environment, testing is often conducted to prove that the workload works in a particular scenario. Testing is not typically used to validate recovery strategies. In the cloud, you can test how your workload fails, and you can validate your recovery procedures. You can use automation to simulate different failures or to recreate scenarios that led to failures before. This approach exposes failure pathways that you can test and fix before a real failure scenario occurs, thus reducing risk.
- **Scale horizontally to increase aggregate workload availability**: Replace one large resource with multiple small resources to reduce the impact of a single failure on the overall workload. Distribute requests across multiple, smaller resources to ensure that they don't share a common point of failure.
- **Stop guessing capacity**: A common cause of failure in on-premises workloads is resource saturation, when the demands placed on a workload exceed the capacity of that workload (this is often the objective of denial of service attacks). In the cloud, you can monitor demand and workload utilization, and automate the addition or removal of resources to maintain the optimal level to satisfy demand without over- or under-provisioning. There are still limits, but some quotas can be controlled and others can be managed (see Manage Service Quotas and Constraints).
- **Manage change in automation**: Changes to your infrastructure should be made using automation. The changes that need to be managed include changes to the automation, which then can be tracked and reviewed.

Definition

There are four best practice areas for reliability in the cloud:

- **Foundations**
- **Workload Architecture**
- **Change Management**
- **Failure Management**

To achieve reliability you must start with the foundations — an environment where service quotas and network topology accommodate the workload. The workload architecture of the distributed system must be designed to prevent and mitigate failures. The workload must handle changes in demand or requirements, and it must be designed to detect failure and automatically heal itself.

Best Practices

Topics

Foundations

Foundational requirements are those whose scope extends beyond a single workload or project. Before architecting any system, foundational requirements that influence reliability should be in place. For example, you must have sufficient network bandwidth to your data center.

With AWS, most of these foundational requirements are already incorporated or can be addressed as needed. The cloud is designed to be nearly limitless, so it's the responsibility of AWS to satisfy the requirement for sufficient networking and compute capacity, leaving you free to change resource size and allocations on demand.

The following questions focus on these considerations for reliability. (For a list of reliability questions and best practices, see the Appendix (p. 58).).

REL 1: How do you manage service quotas and constraints?
For cloud-based workload architectures, there are service quotas (which are also referred to as service limits). These quotas exist to prevent accidentally provisioning more resources than you need and to limit request rates on API operations so as to protect services from abuse. There are also resource constraints, for example, the rate that you can push bits down a fiber-optic cable, or the amount of storage on a physical disk.

REL 2: How do you plan your network topology?
Workloads often exist in multiple environments. These include multiple cloud environments (both publicly accessible and private) and possibly your existing data center infrastructure. Plans must include network considerations such as intra- and inter-system connectivity, public IP address management, private IP address management, and domain name resolution.

For cloud-based workload architectures, there are service quotas (which are also referred to as service limits). These quotas exist to prevent accidentally provisioning more resources than you need and to limit request rates on API operations to protect services from abuse. Workloads often exist in multiple environments. You must monitor and manage these quotas for all workload environments. These include multiple cloud environments (both publicly accessible and private) and may include your existing data center infrastructure. Plans must include network considerations, such as intrasystem and intersystem connectivity, public IP address management, private IP address management, and domain name resolution.

Workload Architecture

A reliable workload starts with upfront design decisions for both software and infrastructure. Your architecture choices will impact your workload behavior across all five Well-Architected pillars. For reliability, there are specific patterns you must follow.

With AWS, workload developers have their choice of languages and technologies to use. AWS SDKs take the complexity out of coding by providing language-specific APIs for AWS services. These SDKs, plus the choice of languages, allow developers to implement the reliability best practices listed here. Developers

can also read about and learn from how Amazon builds and operates software in The Amazon Builders' Library.

The following questions focus on these considerations for reliability.

REL 3: How do you design your workload service architecture?
Build highly scalable and reliable workloads using a service-oriented architecture (SOA) or a microservices architecture. Service-oriented architecture (SOA) is the practice of making software components reusable via service interfaces. Microservices architecture goes further to make components smaller and simpler.

REL 4: How do you design interactions in a distributed system to prevent failures?
Distributed systems rely on communications networks to interconnect components, such as servers or services. Your workload must operate reliably despite data loss or latency in these networks. Components of the distributed system must operate in a way that does not negatively impact other components or the workload. These best practices prevent failures and improve mean time between failures (MTBF).

REL 5: How do you design interactions in a distributed system to mitigate or withstand failures?
Distributed systems rely on communications networks to interconnect components (such as servers or services). Your workload must operate reliably despite data loss or latency over these networks. Components of the distributed system must operate in a way that does not negatively impact other components or the workload. These best practices enable workloads to withstand stresses or failures, more quickly recover from them, and mitigate the impact of such impairments. The result is improved mean time to recovery (MTTR).

Distributed systems rely on communications networks to interconnect components, such as servers or services. Your workload must operate reliably despite data loss or latency in these networks. Components of the distributed system must operate in a way that does not negatively impact other components or the workload.

Change Management

Changes to your workload or its environment must be anticipated and accommodated to achieve reliable operation of the workload. Changes include those imposed on your workload, such as spikes in demand, as well as those from within, such as feature deployments and security patches.

Using AWS, you can monitor the behavior of a workload and automate the response to KPIs. For example, your workload can add additional servers as a workload gains more users. You can control who has permission to make workload changes and audit the history of these changes.

The following questions focus on these considerations for reliability.

REL 6: How do you monitor workload resources?
Logs and metrics are powerful tools to gain insight into the health of your workload. You can configure your workload to monitor logs and metrics and send notifications when thresholds are crossed or significant events occur. Monitoring enables your workload to recognize when low-performance thresholds are crossed or failures occur, so it can recover automatically in response.

REL 7: How do you design your workload to adapt to changes in demand?
A scalable workload provides elasticity to add or remove resources automatically so that they closely match the current demand at any given point in time.

REL 8: How do you implement change?
Controlled changes are necessary to deploy new functionality, and to ensure that the workloads and the operating environment are running known software and can be patched or replaced in a predictable manner. If these changes are uncontrolled, then it makes it difficult to predict the effect of these changes, or to address issues that arise because of them.

When you architect a workload to automatically add and remove resources in response to changes in demand, this not only increases reliability but also ensures that business success doesn't become a burden. With monitoring in place, your team will be automatically alerted when KPIs deviate from expected norms. Automatic logging of changes to your environment allows you to audit and quickly identify actions that might have impacted reliability. Controls on change management ensure that you can enforce the rules that deliver the reliability you need.

Failure Management

In any system of reasonable complexity, it is expected that failures will occur. Reliability requires that your workload be aware of failures as they occur and take action to avoid impact on availability. Workloads must be able to both withstand failures and automatically repair issues.

With AWS, you can take advantage of automation to react to monitoring data. For example, when a particular metric crosses a threshold, you can trigger an automated action to remedy the problem. Also, rather than trying to diagnose and fix a failed resource that is part of your production environment, you can replace it with a new one and carry out the analysis on the failed resource out of band. Since the cloud enables you to stand up temporary versions of a whole system at low cost, you can use automated testing to verify full recovery processes.

The following questions focus on these considerations for reliability.

REL 9: How do you back up data?
Back up data, applications, and configuration to meet your requirements for recovery time objectives (RTO) and recovery point objectives (RPO).

REL 10: How do you use fault isolation to protect your workload?
Fault isolated boundaries limit the effect of a failure within a workload to a limited number of components. Components outside of the boundary are unaffected by the failure. Using multiple fault isolated boundaries, you can limit the impact on your workload.

REL 11: How do you design your workload to withstand component failures?
Workloads with a requirement for high availability and low mean time to recovery (MTTR) must be architected for resiliency.

REL 12: How do you test reliability?
After you have designed your workload to be resilient to the stresses of production, testing is the only way to ensure that it will operate as designed, and deliver the resiliency you expect.

REL 13: How do you plan for disaster recovery (DR)?
Having backups and redundant workload components in place is the start of your DR strategy. RTO and RPO are your objectives for restoration of your workload. Set these based on business needs. Implement a strategy to meet these objectives, considering locations and function of workload resources and data. The probability of disruption and cost of recovery are also key factors that help to inform the business value of providing disaster recovery for a workload.

Regularly back up your data and test your backup files to ensure that you can recover from both logical and physical errors. A key to managing failure is the frequent and automated testing of workloads to cause failure, and then observe how they recover. Do this on a regular schedule and ensure that such testing is also triggered after significant workload changes. Actively track KPIs, as well as the recovery time objective (RTO) and recovery point objective (RPO), to assess a workload's resiliency (especially under failure-testing scenarios). Tracking KPIs will help you identify and mitigate single points of failure. The objective is to thoroughly test your workload-recovery processes so that you are confident that you can recover all your data and continue to serve your customers, even in the face of sustained problems. Your recovery processes should be as well exercised as your normal production processes.

Resources

Refer to the following resources to learn more about our best practices for Reliability.

Documentation

- AWS Documentation
- AWS Global Infrastructure
- AWS Auto Scaling: How Scaling Plans Work
- What Is AWS Backup?

Whitepaper

- Reliability Pillar: AWS Well-Architected
- Implementing Microservices on AWS

Performance Efficiency

The Performance Efficiency pillar includes the ability to use computing resources efficiently to meet system requirements, and to maintain that efficiency as demand changes and technologies evolve.

The performance efficiency pillar provides an overview of design principles, best practices, and questions. You can find prescriptive guidance on implementation in the Performance Efficiency Pillar whitepaper.

Topics

Design Principles

There are five design principles for performance efficiency in the cloud:

- **Democratize advanced technologies**: Make advanced technology implementation easier for your team by delegating complex tasks to your cloud vendor. Rather than asking your IT team to learn about hosting and running a new technology, consider consuming the technology as a service. For example, NoSQL databases, media transcoding, and machine learning are all technologies that require specialized expertise. In the cloud, these technologies become services that your team can consume, allowing your team to focus on product development rather than resource provisioning and management.
- **Go global in minutes**: Deploying your workload in multiple AWS Regions around the world allows you to provide lower latency and a better experience for your customers at minimal cost.
- **Use serverless architectures**: Serverless architectures remove the need for you to run and maintain physical servers for traditional compute activities. For example, serverless storage services can act as static websites (removing the need for web servers) and event services can host code. This removes the operational burden of managing physical servers, and can lower transactional costs because managed services operate at cloud scale.
- **Experiment more often**: With virtual and automatable resources, you can quickly carry out comparative testing using different types of instances, storage, or configurations.
- **Consider mechanical sympathy**: Understand how cloud services are consumed and always use the technology approach that aligns best with your workload goals. For example, consider data access patterns when you select database or storage approaches.

Definition

There are four best practice areas for performance efficiency in the cloud:

- **Selection**
- **Review**
- **Monitoring**
- **Tradeoffs**

Take a data-driven approach to building a high-performance architecture. Gather data on all aspects of the architecture, from the high-level design to the selection and configuration of resource types.

Reviewing your choices on a regular basis ensures that you are taking advantage of the continually evolving AWS Cloud. Monitoring ensures that you are aware of any deviance from expected performance. Make trade-offs in your architecture to improve performance, such as using compression or caching, or relaxing consistency requirements.

Best Practices

Topics

Selection

The optimal solution for a particular workload varies, and solutions often combine multiple approaches. Well-architected workloads use multiple solutions and enable different features to improve performance.

AWS resources are available in many types and configurations, which makes it easier to find an approach that closely matches your workload needs. You can also find options that are not easily achievable with on-premises infrastructure. For example, a managed service such as Amazon DynamoDB provides a fully managed NoSQL database with single-digit millisecond latency at any scale.

The following question focuses on these considerations for performance efficiency. (For a list of performance efficiency questions and best practices, see the Appendix (p. 65).).

PERF 1: How do you select the best performing architecture?
Often, multiple approaches are required for optimal performance across a workload. Well-architected systems use multiple solutions and features to improve performance.

Use a data-driven approach to select the patterns and implementation for your architecture and achieve a cost effective solution. AWS Solutions Architects, AWS Reference Architectures, and AWS Partner Network (APN) partners can help you select an architecture based on industry knowledge, but data obtained through benchmarking or load testing will be required to optimize your architecture.

Your architecture will likely combine a number of different architectural approaches (for example, event-driven, ETL, or pipeline). The implementation of your architecture will use the AWS services that are specific to the optimization of your architecture's performance. In the following sections we discuss the four main resource types to consider (compute, storage, database, and network).

Compute

Selecting compute resources that meet your requirements, performance needs, and provide great efficiency of cost and effort will enable you to accomplish more with the same number of resources. When evaluating compute options, be aware of your requirements for workload performance and cost requirements and use this to make informed decisions.

In AWS, compute is available in three forms: instances, containers, and functions:

- **Instances** are virtualized servers, allowing you to change their capabilities with a button or an API call. Because resource decisions in the cloud aren't fixed, you can experiment with different server types. At AWS, these virtual server instances come in different families and sizes, and they offer a wide variety of capabilities, including solid-state drives (SSDs) and graphics processing units (GPUs).
- **Containers** are a method of operating system virtualization that allow you to run an application and its dependencies in resource-isolated processes. AWS Fargate is serverless compute for containers or Amazon EC2 can be used if you need control over the installation, configuration, and management of your compute environment. You can also choose from multiple container orchestration platforms: Amazon Elastic Container Service (ECS) or Amazon Elastic Kubernetes Service (EKS).
- **Functions** abstract the execution environment from the code you want to execute. For example, AWS Lambda allows you to execute code without running an instance.

The following question focuses on these considerations for performance efficiency.

PERF 2: How do you select your compute solution?
The optimal compute solution for a workload varies based on application design, usage patterns, and configuration settings. Architectures can use different compute solutions for various components and enable different features to improve performance. Selecting the wrong compute solution for an architecture can lead to lower performance efficiency.

When architecting your use of compute you should take advantage of the elasticity mechanisms available to ensure you have sufficient capacity to sustain performance as demand changes.

Storage

Cloud storage is a critical component of cloud computing, holding the information used by your workload. Cloud storage is typically more reliable, scalable, and secure than traditional on-premises storage systems. Select from object, block, and file storage services as well as cloud data migration options for your workload.

In AWS, storage is available in three forms: object, block, and file:

- **Object Storage** provides a scalable, durable platform to make data accessible from any internet location for user-generated content, active archive, serverless computing, Big Data storage or backup and recovery. Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. Amazon S3 is designed for 99.999999999% (11 9's) of durability, and stores data for millions of applications for companies all around the world.
- **Block Storage** provides highly available, consistent, low-latency block storage for each virtual host and is analogous to direct-attached storage (DAS) or a Storage Area Network (SAN). Amazon Elastic Block Store (Amazon EBS) is designed for workloads that require persistent storage accessible by EC2 instances that helps you tune applications with the right storage capacity, performance and cost.
- **File Storage** provides access to a shared file system across multiple systems. File storage solutions like Amazon Elastic File System (EFS) are ideal for use cases, such as large content repositories, development environments, media stores, or user home directories. Amazon FSx makes it easy and cost effective to launch and run popular file systems so you can leverage the rich feature sets and fast performance of widely used open source and commercially-licensed file systems.

The following question focuses on these considerations for performance efficiency.

PERF 3: How do you select your storage solution?
The optimal storage solution for a system varies based on the kind of access method (block, file, or object), patterns of access (random or sequential), required throughput, frequency of access (online, offline, archival), frequency of update (WORM, dynamic), and availability and durability constraints. Well-architected systems use multiple storage solutions and enable different features to improve performance and use resources efficiently.

When you select a storage solution, ensuring that it aligns with your access patterns will be critical to achieving the performance you want.

Database

The cloud offers purpose-built database services that address different problems presented by your workload. You can choose from many purpose-built database engines including relational, key-value, document, in-memory, graph, time series, and ledger databases. By picking the best database to

solve a specific problem (or a group of problems), you can break away from restrictive one-size-fits-all monolithic databases and focus on building applications to meet the performance needs of your customers.

In AWS you can choose from multiple purpose-built database engines including relational, key-value, document, in-memory, graph, time series, and ledger databases. With AWS databases, you don't need to worry about database management tasks such as server provisioning, patching, setup, configuration, backups, or recovery. AWS continuously monitors your clusters to keep your workloads up and running with self-healing storage and automated scaling, so that you can focus on higher value application development.

The following question focuses on these considerations for performance efficiency.

PERF 4: How do you select your database solution?
The optimal database solution for a system varies based on requirements for availability, consistency, partition tolerance, latency, durability, scalability, and query capability. Many systems use different database solutions for various subsystems and enable different features to improve performance. Selecting the wrong database solution and features for a system can lead to lower performance efficiency.

Your workload's database approach has a significant impact on performance efficiency. It's often an area that is chosen according to organizational defaults rather than through a data-driven approach. As with storage, it is critical to consider the access patterns of your workload, and also to consider if other non-database solutions could solve the problem more efficiently (such as using graph, time series, or in-memory storage database).

Network

Since the network is between all workload components, it can have great impacts, both positive and negative, on workload performance and behavior. There are also workloads that are heavily dependent on network performance such as High Performance Computing (HPC) where deep network understanding is important to increase cluster performance. You must determine the workload requirements for bandwidth, latency, jitter, and throughput.

On AWS, networking is virtualized and is available in a number of different types and configurations. This makes it easier to match your networking methods with your needs. AWS offers product features (for example, Enhanced Networking, Amazon EBS-optimized instances, Amazon S3 transfer acceleration, and dynamic Amazon CloudFront) to optimize network traffic. AWS also offers networking features (for example, Amazon Route 53 latency routing, Amazon VPC endpoints, AWS Direct Connect, and AWS Global Accelerator) to reduce network distance or jitter.

The following question focuses on these considerations for performance efficiency.

PERF 5: How do you configure your networking solution?
The optimal network solution for a workload varies based on latency, throughput requirements, jitter, and bandwidth. Physical constraints, such as user or on-premises resources, determine location options. These constraints can be offset with edge locations or resource placement.

You must consider location when deploying your network. You can choose to place resources close to where they will be used to reduce distance. Use networking metrics to make changes to networking configuration as the workload evolves. By taking advantage of Regions, placement groups, and edge services, you can significantly improve performance. Cloud based networks can be quickly re-built

or modified, so evolving your network architecture over time is necessary to maintain performance efficiency.

Review

Cloud technologies are rapidly evolving and you must ensure that workload components are using the latest technologies and approaches to continually improve performance. You must continually evaluate and consider changes to your workload components to ensure you are meeting its performance and cost objectives. New technologies, such as machine learning and artificial intelligence (AI), can allow you to reimagine customer experiences and innovate across all of your business workloads.

Take advantage of the continual innovation at AWS driven by customer need. We release new Regions, edge locations, services, and features regularly. Any of these releases could positively improve the performance efficiency of your architecture.

The following question focuses on these considerations for performance efficiency.

PERF 6: How do you evolve your workload to take advantage of new releases?
When architecting workloads, there are finite options that you can choose from. However, over time, new technologies and approaches become available that could improve the performance of your workload.

Architectures performing poorly are usually the result of a non-existent or broken performance review process. If your architecture is performing poorly, implementing a performance review process will allow you to apply Deming's plan-do-check-act (PDCA) cycle to drive iterative improvement.

Monitoring

After you implement your workload, you must monitor its performance so that you can remediate any issues before they impact your customers. Monitoring metrics should be used to raise alarms when thresholds are breached.

Amazon CloudWatch is a monitoring and observability service that provides you with data and actionable insights to monitor your workload, respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health. CloudWatch collects monitoring and operational data in the form of logs, metrics, and events from workloads that run on AWS and on-premises servers. AWS X-Ray helps developers analyze and debug production, distributed applications. With AWS X-Ray, you can glean insights into how your application is performing and discover root causes and identify performance bottlenecks. You can use these insights to react quickly and keep your workload running smoothly.

The following question focuses on these considerations for performance efficiency.

PERF 7: How do you monitor your resources to ensure they are performing?
System performance can degrade over time. Monitor system performance to identify degradation and remediate internal or external factors, such as the operating system or application load.

Ensuring that you do not see false positives is key to an effective monitoring solution. Automated triggers avoid human error and can reduce the time it takes to fix problems. Plan for game days, where simulations are conducted in the production environment, to test your alarm solution and ensure that it correctly recognizes issues.

Tradeoffs

When you architect solutions, think about tradeoffs to ensure an optimal approach. Depending on your situation, you could trade consistency, durability, and space for time or latency, to deliver higher performance.

Using AWS, you can go global in minutes and deploy resources in multiple locations across the globe to be closer to your end users. You can also dynamically add readonly replicas to information stores (such as database systems) to reduce the load on the primary database.

The following question focuses on these considerations for performance efficiency.

PERF 8: How do you use tradeoffs to improve performance?
When architecting solutions, determining tradeoffs enables you to select an optimal approach. Often you can improve performance by trading consistency, durability, and space for time and latency.

As you make changes to the workload, collect and evaluate metrics to determine the impact of those changes. Measure the impacts to the system and to the end-user to understand how your trade-offs impact your workload. Use a systematic approach, such as load testing, to explore whether the tradeoff improves performance.

Resources

Refer to the following resources to learn more about our best practices for Performance Efficiency.

Documentation

- Amazon S3 Performance Optimization
- Amazon EBS Volume Performance

Whitepaper

- Performance Efficiency Pillar

Video

- AWS re:Invent 2019: Amazon EC2 foundations (CMP211-R2)
- AWS re:Invent 2019: Leadership session: Storage state of the union (STG201-L)
- AWS re:Invent 2019: Leadership session: AWS purpose-built databases (DAT209-L)
- AWS re:Invent 2019: Connectivity to AWS and hybrid AWS network architectures (NET317-R1)
- AWS re:Invent 2019: Powering next-gen Amazon EC2: Deep dive into the Nitro system (CMP303-R2)
- AWS re:Invent 2019: Scaling up to your first 10 million users (ARC211-R)

Cost Optimization

The Cost Optimization pillar includes the ability to run systems to deliver business value at the lowest price point.

The cost optimization pillar provides an overview of design principles, best practices, and questions. You can find prescriptive guidance on implementation in the Cost Optimization Pillar whitepaper.

Topics

Design Principles

There are five design principles for cost optimization in the cloud:

- **Implement Cloud Financial Management**: To achieve financial success and accelerate business value realization in the cloud, you need to invest in Cloud Financial Management /Cost Optimization. Your organization needs to dedicate time and resources to build capability in this new domain of technology and usage management. Similar to your Security or Operational Excellence capability, you need to build capability through knowledge building, programs, resources, and processes to become a cost-efficient organization.
- **Adopt a consumption model**: Pay only for the computing resources that you require and increase or decrease usage depending on business requirements, not by using elaborate forecasting. For example, development and test environments are typically only used for eight hours a day during the work week. You can stop these resources when they are not in use for a potential cost savings of 75% (40 hours versus 168 hours).
- **Measure overall efficiency**: Measure the business output of the workload and the costs associated with delivering it. Use this measure to know the gains you make from increasing output and reducing costs.
- **Stop spending money on undifferentiated heavy lifting**: AWS does the heavy lifting of data center operations like racking, stacking, and powering servers. It also removes the operational burden of managing operating systems and applications with managed services. This allows you to focus on your customers and business projects rather than on IT infrastructure.
- **Analyze and attribute expenditure**: The cloud makes it easier to accurately identify the usage and cost of systems, which then allows transparent attribution of IT costs to individual workload owners. This helps measure return on investment (ROI) and gives workload owners an opportunity to optimize their resources and reduce costs.

Definition

There are five best practice areas for cost optimization in the cloud:

- **Practice Cloud Financial Management**
- **Expenditure and usage awareness**
- **Cost-effective resources**
- **Manage demand and supply resources**
- **Optimize over time**

As with the other pillars within the Well-Architected Framework, there are tradeoffs to consider, for example, whether to optimize for speed-to-market or for cost. In some cases, it's best to optimize for speed—going to market quickly, shipping new features, or simply meeting a deadline—rather than investing in up-front cost optimization. Design decisions are sometimes directed by haste rather than data, and the temptation always exists to overcompensate "just in case" rather than spend time benchmarking for the most cost-optimal deployment. This might lead to over-provisioned and under-optimized deployments. However, this is a reasonable choice when you need to "lift and shift" resources from your on-premises environment to the cloud and then optimize afterwards. Investing the right

amount of effort in a cost optimization strategy up front allows you to realize the economic benefits of the cloud more readily by ensuring a consistent adherence to best practices and avoiding unnecessary over provisioning. The following sections provide techniques and best practices for both the initial and ongoing implementation of Cloud Financial Management and cost optimization of your workloads.

Best Practices

Topics

Practice Cloud Financial Management

With the adoption of cloud, technology teams innovate faster due to shortened approval, procurement, and infrastructure deployment cycles. A new approach to financial management in the cloud is required to realize business value and financial success. This approach is Cloud Financial Management, and builds capability across your organization by implementing organizational wide knowledge building, programs, resources, and processes.

Many organizations are composed of many different units with different priorities. The ability to align your organization to an agreed set of financial objectives, and provide your organization the mechanisms to meet them, will create a more efficient organization. A capable organization will innovate and build faster, be more agile and adjust to any internal or external factors.

In AWS you can use Cost Explorer, and optionally Amazon Athena and Amazon QuickSight with the Cost and Usage Report (CUR), to provide cost and usage awareness throughout your organization. AWS Budgets provides proactive notifications for cost and usage. The AWS blogs provide information on new services and features to ensure you keep up to date with new service releases.

The following question focuses on these considerations for cost optimization. (For a list of cost optimization questions and best practices, see the Appendix (p. 70).).

COST 1: How do you implement cloud financial management?
Implementing Cloud Financial Management enables organizations to realize business value and financial success as they optimize their cost and usage and scale on AWS.

When building a cost optimization function, use members and supplement the team with experts in CFM and CO. Existing team members will understand how the organization currently functions and how to rapidly implement improvements. Also consider including people with supplementary or specialist skill sets, such as analytics and project management.

When implementing cost awareness in your organization, improve or build on existing programs and processes. It is much faster to add to what exists than to build new processes and programs. This will result in achieving outcomes much faster.

Expenditure and usage awareness

The increased flexibility and agility that the cloud enables encourages innovation and fast-paced development and deployment. It eliminates the manual processes and time associated with provisioning

on-premises infrastructure, including identifying hardware specifications, negotiating price quotations, managing purchase orders, scheduling shipments, and then deploying the resources. However, the ease of use and virtually unlimited on-demand capacity requires a new way of thinking about expenditures.

Many businesses are composed of multiple systems run by various teams. The capability to attribute resource costs to the individual organization or product owners drives efficient usage behavior and helps reduce waste. Accurate cost attribution allows you to know which products are truly profitable, and allows you to make more informed decisions about where to allocate budget.

In AWS, you create an account structure with AWS Organizations or AWS Control Tower, which provides separation and assists in allocation of your costs and usage. You can also use resource tagging to apply business and organization information to your usage and cost. Use AWS Cost Explorer for visibility into your cost and usage, or create customized dashboards and analytics with Amazon Athena and Amazon QuickSight. Controlling your cost and usage is done by notifications through AWS Budgets, and controls using AWS Identity and Access Management (IAM), and Service Quotas.

The following questions focus on these considerations for cost optimization.

COST 2: How do you govern usage?
Establish policies and mechanisms to ensure that appropriate costs are incurred while objectives are achieved. By employing a checks-and-balances approach, you can innovate without overspending.

COST 3: How do you monitor usage and cost?
Establish policies and procedures to monitor and appropriately allocate your costs. This allows you to measure and improve the cost efficiency of this workload.

COST 4: How do you decommission resources?
Implement change control and resource management from project inception to end-of-life. This ensures you shut down or terminate unused resources to reduce waste.

You can use cost allocation tags to categorize and track your AWS usage and costs. When you apply tags to your AWS resources (such as EC2 instances or S3 buckets), AWS generates a cost and usage report with your usage and your tags. You can apply tags that represent organization categories (such as cost centers, workload names, or owners) to organize your costs across multiple services.

Ensure you use the right level of detail and granularity in cost and usage reporting and monitoring. For high level insights and trends, use daily granularity with AWS Cost Explorer. For deeper analysis and inspection use hourly granularity in AWS Cost Explorer, or Amazon Athena and Amazon QuickSight with the Cost and Usage Report (CUR) at an hourly granularity.

Combining tagged resources with entity lifecycle tracking (employees, projects) makes it possible to identify orphaned resources or projects that are no longer generating value to the organization and should be decommissioned. You can set up billing alerts to notify you of predicted overspending.

Cost-effective resources

Using the appropriate instances and resources for your workload is key to cost savings. For example, a reporting process might take five hours to run on a smaller server but one hour to run on a larger server that is twice as expensive. Both servers give you the same outcome, but the smaller server incurs more cost over time.

A well-architected workload uses the most cost-effective resources, which can have a significant and positive economic impact. You also have the opportunity to use managed services to reduce costs. For example, rather than maintaining servers to deliver email, you can use a service that charges on a per-message basis.

AWS offers a variety of flexible and cost-effective pricing options to acquire instances from Amazon EC2 and other services in a way that best fits your needs. *On-Demand Instances* allow you to pay for compute capacity by the hour, with no minimum commitments required. *Savings Plans and Reserved Instances* offer savings of up to 75% off On-Demand pricing. With Spot Instances, you can leverage unused Amazon EC2 capacity and offer savings of up to 90% off On-Demand pricing. *Spot Instances* are appropriate where the system can tolerate using a fleet of servers where individual servers can come and go dynamically, such as stateless web servers, batch processing, or when using HPC and big data.

Appropriate service selection can also reduce usage and costs; such as CloudFront to minimize data transfer, or completely eliminate costs, such as utilizing Amazon Aurora on RDS to remove expensive database licensing costs.

The following questions focus on these considerations for cost optimization.

COST 5: How do you evaluate cost when you select services?
Amazon EC2, Amazon EBS, and Amazon S3 are building-block AWS services. Managed services, such as Amazon RDS and Amazon DynamoDB, are higher level, or application level, AWS services. By selecting the appropriate building blocks and managed services, you can optimize this workload for cost. For example, using managed services, you can reduce or remove much of your administrative and operational overhead, freeing you to work on applications and business-related activities.

COST 6: How do you meet cost targets when you select resource type, size and number?
Ensure that you choose the appropriate resource size and number of resources for the task at hand. You minimize waste by selecting the most cost effective type, size, and number.

COST 7: How do you use pricing models to reduce cost?
Use the pricing model that is most appropriate for your resources to minimize expense.

COST 8: How do you plan for data transfer charges?
Ensure that you plan and monitor data transfer charges so that you can make architectural decisions to minimize costs. A small yet effective architectural change can drastically reduce your operational costs over time.

By factoring in cost during service selection, and using tools such as Cost Explorer and AWS Trusted Advisor to regularly review your AWS usage, you can actively monitor your utilization and adjust your deployments accordingly.

Manage demand and supply resources

When you move to the cloud, you pay only for what you need. You can supply resources to match the workload demand at the time they're needed, this eliminates the need for costly and wasteful over

provisioning. You can also modify the demand, using a throttle, buffer, or queue to smooth the demand and serve it with less resources resulting in a lower cost, or process it at a later time with a batch service.

In AWS, you can automatically provision resources to match the workload demand. Auto Scaling using demand or time-based approaches allow you to add and remove resources as needed. If you can anticipate changes in demand, you can save more money and ensure your resources match your workload needs. You can use Amazon API Gateway to implement throttling, or Amazon SQS to implementing a queue in your workload. These will both allow you to modify the demand on your workload components.

The following question focuses on these considerations for cost optimization.

COST 9: How do you manage demand, and supply resources?
For a workload that has balanced spend and performance, ensure that everything you pay for is used and avoid significantly underutilizing instances. A skewed utilization metric in either direction has an adverse impact on your organization, in either operational costs (degraded performance due to over-utilization), or wasted AWS expenditures (due to over-provisioning).

When designing to modify demand and supply resources, actively think about the patterns of usage, the time it takes to provision new resources, and the predictability of the demand pattern. When managing demand, ensure you have a correctly sized queue or buffer, and that you are responding to workload demand in the required amount of time.

Optimize over time

As AWS releases new services and features, it's a best practice to review your existing architectural decisions to ensure they continue to be the most cost effective. As your requirements change, be aggressive in decommissioning resources, entire services, and systems that you no longer require.

Implementing new features or resource types can optimize your workload incrementally, while minimizing the effort required to implement the change. This provides continual improvements in efficiency over time and ensures you remain on the most updated technology to reduce operating costs. You can also replace or add new components to the workload with new services. This can provide significant increases in efficiency, so it's essential to regularly review your workload, and implement new services and features.

The following question focuses on these considerations for cost optimization.

COST 10: How do you evaluate new services?
As AWS releases new services and features, it's a best practice to review your existing architectural decisions to ensure they continue to be the most cost effective.

When regularly reviewing your deployments, assess how newer services can help save you money. For example, Amazon Aurora on RDS can reduce costs for relational databases. Using serverless such as Lambda can remove the need to operate and manage instances to run code.

Resources

Refer to the following resources to learn more about our best practices for Cost Optimization.

Documentation

- AWS Documentation

Whitepaper

- Cost Optimization Pillar

AWS Well-Architected Framework

The Review Process

The review of architectures needs to be done in a consistent manner, with a blamefree approach that encourages diving deep. It should be a light weight process (hours not days) that is a conversation and not an audit. The purpose of reviewing an architecture is to identify any critical issues that might need addressing or areas that could be improved. The outcome of the review is a set of actions that should improve the experience of a customer using the workload.

As discussed in the “On Architecture” section, you will want each team member to take responsibility for the quality of its architecture. We recommend that the team members who build an architecture use the Well-Architected Framework to continually review their architecture, rather than holding a formal review meeting. A continuous approach allows your team members to update answers as the architecture evolves, and improve the architecture as you deliver features.

The AWS Well-Architected Framework is aligned to the way that AWS reviews systems and services internally. It is premised on a set of design principles that influences architectural approach, and questions that ensure that people don’t neglect areas that often featured in Root Cause Analysis (RCA). Whenever there is a significant issue with an internal system, AWS service, or customer, we look at the RCA to see if we could improve the review processes we use.

Reviews should be applied at key milestones in the product lifecycle, early on in the design phase to avoid *one-way doors* that are difficult to change, and then before the go-live date. After you go into production, your workload will continue to evolve as you add new features and change technology implementations. The architecture of a workload changes over time. You will need to follow good hygiene practices to stop its architectural characteristics from degrading as you evolve it. As you make significant architecture changes you should follow a set of hygiene processes including a Well-Architected review.

If you want to use the review as a one-time snapshot or independent measurement, you will want to ensure that you have all the right people in the conversation. Often, we find that reviews are the first time that a team truly understands what they have implemented. An approach that works well when reviewing another team's workload is to have a series of informal conversations about their architecture where you can glean the answers to most questions. You can then follow up with one or two meetings where you can gain clarity or dive deep on areas of ambiguity or perceived risk.

Here are some suggested items to facilitate your meetings:

- A meeting room with whiteboards
- Print outs of any diagrams or design notes
- Action list of questions that require out-of-band research to answer (for example, “did we enable encryption or not?”)

After you have done a review, you should have a list of issues that you can prioritize based on your business context. You will also want to take into account the impact of those issues on the day-to-day work of your team. If you address these issues early, you could free up time to work on creating business value rather than solving recurring problems. As you address issues, you can update your review to see how the architecture is improving.

While the value of a review is clear after you have done one, you may find that a new team might be resistant at first. Here are some objections that can be handled through educating the team on the benefits of a review:

- “We are too busy!” (Often said when the team is getting ready for a big launch.)

    - If you are getting ready for a big launch you will want it to go smoothly. The review will allow you to understand any problems you might have missed.
    - We recommend that you carry out reviews early in the product lifecycle to uncover risks and develop a mitigation plan aligned with the feature delivery roadmap.
- "We don't have time to do anything with the results!" (Often said when there is an immovable event, such as the Super Bowl, that they are targeting.)
    - These events can't be moved. Do you really want to go into it without knowing the risks in your architecture? Even if you don't address all of these issues you can still have playbooks for handling them if they materialize.
- "We don't want others to know the secrets of our solution implementation!"
    - If you point the team at the questions in the Well-Architected Framework, they will see that none of the questions reveal any commercial or technical propriety information.

As you carry out multiple reviews with teams in your organization, you might identify thematic issues. For example, you might see that a group of teams has clusters of issues in a particular pillar or topic. You will want to look at all your reviews in a holistic manner, and identify any mechanisms, training, or principal engineering talks that could help address those thematic issues.

AWS Well-Architected Framework

Conclusion

The AWS Well-Architected Framework provides architectural best practices across the five pillars for designing and operating reliable, secure, efficient, and cost-effective systems in the cloud. The Framework provides a set of questions that allows you to review an existing or proposed architecture. It also provides a set of AWS best practices for each pillar. Using the Framework in your architecture will help you produce stable and efficient systems, which allow you to focus on your functional requirements.

AWS Well-Architected Framework

Contributors

The following individuals and organizations contributed to this document:

- Rodney Lester: Senior Manager Well-Architected, Amazon Web Services
- Brian Carlson: Operations Lead Well-Architected, Amazon Web Services
- Ben Potter: Security Lead Well-Architected, Amazon Web Services
- Eric Pullen: Performance Lead Well-Architected, Amazon Web Services
- Seth Eliot: Reliability Lead Well-Architected, Amazon Web Services
- Nathan Besh: Cost Lead Well-Architected, Amazon Web Services
- Jon Steele: Sr. Technical Account Manager, Amazon Web Services
- Ryan King: Technical Program Manager, Amazon Web Services
- Erin Rifkin: Senior Product Manager, Amazon Web Services
- Max Ramsay: Principal Security Solutions Architect, Amazon Web Services
- Scott Paddock: Security Solutions Architect, Amazon Web Services
- Callum Hughes: Solutions Architect, Amazon Web Services
- Philip Fitzsimons, Sr Manager Well-Architected, Amazon Web Services

Further Reading

AWS Cloud Compliance

AWS Well-Architected Partner program

AWS Well-Architected Tool

AWS Well-Architected homepage

Cost Optimization Pillar whitepaper

Operational Excellence Pillar whitepaper

Performance Efficiency Pillar whitepaper

Reliability Pillar whitepaper

Security Pillar whitepaper

The Amazon Builders' Library

Document Revisions

To be notified about updates to this whitepaper, subscribe to the RSS feed.

update-history-change	update-history-description	update-history-date
Minor update (p. 42)	Fixed numerous links.	March 10, 2021
Minor update (p. 42)	Minor editorial changes throughout.	July 15, 2020
Updates for new Framework (p. 42)	Review and rewrite of most questions and answers.	July 8, 2020
Whitepaper updated (p. 42)	Addition of AWS Well-Architected Tool, links to AWS Well-Architected Labs, and AWS Well-Architected Partners, minor fixes to enable multiple language version of framework.	July 1, 2019
Whitepaper updated (p. 42)	Review and rewrite of most questions and answers, to ensure questions focus on one topic at a time. This caused some previous questions to be split into multiple questions. Added common terms to definitions (workload, component etc). Changed presentation of question in main body to include descriptive text.	November 1, 2018
Whitepaper updated (p. 42)	Updates to simplify question text, standardize answers, and improve readability.	June 1, 2018
Whitepaper updated (p. 42)	Operational Excellence moved to front of pillars and rewritten so it frames other pillars. Refreshed other pillars to reflect evolution of AWS.	November 1, 2017
Whitepaper updated (p. 42)	Updated the Framework to include operational excellence pillar, and revised and updated the other pillars to reduce duplication and incorporate learnings from carrying out reviews with thousands of customers.	November 1, 2016
Minor updates (p. 42)	Updated the Appendix with current Amazon CloudWatch Logs information.	November 1, 2015

AWS Well-Architected Framework

Initial publication (p. 42)	AWS Well-Architected Framework published.	October 1, 2015

Appendix: Questions and Best Practices

Topics

Operational Excellence

Topics

Organization

OPS 1 How do you determine what your priorities are?

Everyone needs to understand their part in enabling business success. Have shared goals in order to set priorities for resources. This will maximize the benefits of your efforts.

Best Practices:

- **Evaluate external customer needs**: Involve key stakeholders, including business, development, and operations teams, to determine where to focus efforts on external customer needs. This will ensure that you have a thorough understanding of the operations support that is required to achieve your desired business outcomes.
- **Evaluate internal customer needs**: Involve key stakeholders, including business, development, and operations teams, when determining where to focus efforts on internal customer needs. This will ensure that you have a thorough understanding of the operations support that is required to achieve business outcomes.
- **Evaluate governance requirements**: Ensure that you are aware of guidelines or obligations defined by your organization that may mandate or emphasize specific focus. Evaluate internal factors, such as organization policy, standards, and requirements. Validate that you have mechanisms to identify changes to governance. If no governance requirements are identified, ensure that you have applied due diligence to this determination.
- **Evaluate compliance requirements**: Evaluate external factors, such as regulatory compliance requirements and industry standards, to ensure that you are aware of guidelines or obligations that

may mandate or emphasize specific focus. If no compliance requirements are identified, ensure that you apply due diligence to this determination.

- **Evaluate threat landscape**: Evaluate threats to the business (for example, competition, business risk and liabilities, operational risks, and information security threats) and maintain current information in a risk registry. Include the impact of risks when determining where to focus efforts.
- **Evaluate tradeoffs**: Evaluate the impact of tradeoffs between competing interests or alternative approaches, to help make informed decisions when determining where to focus efforts or choosing a course of action. For example, accelerating speed to market for new features may be emphasized over cost optimization, or you may choose a relational database for non-relational data to simplify the effort to migrate a system, rather than migrating to a database optimized for your data type and updating your application.
- **Manage benefits and risks**: Manage benefits and risks to make informed decisions when determining where to focus efforts. For example, it may be beneficial to deploy a workload with unresolved issues so that significant new features can be made available to customers. It may be possible to mitigate associated risks, or it may become unacceptable to allow a risk to remain, in which case you will take action to address the risk.

OPS 2 How do you structure your organization to support your business outcomes?

Your teams must understand their part in achieving business outcomes. Teams need to understand their roles in the success of other teams, the role of other teams in their success, and have shared goals. Understanding responsibility, ownership, how decisions are made, and who has authority to make decisions will help focus efforts and maximize the benefits from your teams.

Best Practices:

- **Resources have identified owners**: Understand who has ownership of each application, workload, platform, and infrastructure component, what business value is provided by that component, and why that ownership exists. Understanding the business value of these individual components and how they support business outcomes informs the processes and procedures applied against them.
- **Processes and procedures have identified owners**: Understand who has ownership of the definition of individual processes and procedures, why those specific process and procedures are used, and why that ownership exists. Understanding the reasons that specific processes and procedures are used enables identification of improvement opportunities.
- **Operations activities have identified owners responsible for their performance**: Understand who has responsibility to perform specific activities on defined workloads and why that responsibility exists. Understanding who has responsibility to perform activities informs who will conduct the activity, validate the result, and provide feedback to the owner of the activity.
- **Team members know what they are responsible for**: Understanding the responsibilities of your role and how you contribute to business outcomes informs the prioritization of your tasks and why your role is important. This enables team members to recognize needs and respond appropriately.
- **Mechanisms exist to identify responsibility and ownership**: Where no individual or team is identified, there are defined escalation paths to someone with the authority to assign ownership or plan for that need to be addressed.
- **Mechanisms exist to request additions, changes, and exceptions**: You are able to make requests to owners of processes, procedures, and resources. Make informed decisions to approve requests where viable and determined to be appropriate after an evaluation of benefits and risks.
- **Responsibilities between teams are predefined or negotiated**: There are defined or negotiated agreements between teams describing how they work with and support each other (for example, response times, service level objectives, or service level agreements). Understanding the impact of the teams' work on business outcomes, and the outcomes of other teams and organizations, informs the prioritization of their tasks and enables them to respond appropriately.

OPS 3 How does your organizational culture support your business outcomes?

Provide support for your team members so that they can be more effective in taking action and supporting your business outcome.

Best Practices:

- **Executive Sponsorship**: Senior leadership clearly sets expectations for the organization and evaluates success. Senior leadership is the sponsor, advocate, and driver for the adoption of best practices and evolution of the organization
- **Team members are empowered to take action when outcomes are at risk**: The workload owner has defined guidance and scope empowering team members to respond when outcomes are at risk. Escalation mechanisms are used to get direction when events are outside of the defined scope.
- **Escalation is encouraged**: Team members have mechanisms and are encouraged to escalate concerns to decision makers and stakeholders if they believe outcomes are at risk. Escalation should be performed early and often so that risks can be identified, and prevented from causing incidents.
- **Communications are timely, clear, and actionable**: Mechanisms exist and are used to provide timely notice to team members of known risks and planned events. Necessary context, details, and time (when possible) are provided to support determining if action is necessary, what action is required, and to take action in a timely manner. For example, providing notice of software vulnerabilities so that patching can be expedited, or providing notice of planned sales promotions so that a change freeze can be implemented to avoid the risk of service disruption.
- **Experimentation is encouraged**: Experimentation accelerates learning and keeps team members interested and engaged. An undesired result is a successful experiment that has identified a path that will not lead to success. Team members are not punished for successful experiments with undesired results. Experimentation is required for innovation to happen and turn ideas into outcomes.
- **Team members are enabled and encouraged to maintain and grow their skill sets**: Teams must grow their skill sets to adopt new technologies, and to support changes in demand and responsibilities in support of your workloads. Growth of skills in new technologies is frequently a source of team member satisfaction and supports innovation. Support your team members' pursuit and maintenance of industry certifications that validate and acknowledge their growing skills. Cross train to promote knowledge transfer and reduce the risk of significant impact when you lose skilled and experienced team members with institutional knowledge. Provide dedicated structured time for learning.
- **Resource teams appropriately**: Maintain team member capacity, and provide tools and resources, to support your workload needs. Overtasking team members increases the risk of incidents resulting from human error. Investments in tools and resources (for example, providing automation for frequently executed activities) can scale the effectiveness of your team, enabling them to support additional activities.
- **Diverse opinions are encouraged and sought within and across teams**: Leverage cross-organizational diversity to seek multiple unique perspectives. Use this perspective to increase innovation, challenge your assumptions, and reduce the risk of confirmation bias. Grow inclusion, diversity, and accessibility within your teams to gain beneficial perspectives.

Prepare

OPS 4 How do you design your workload so that you can understand its state?

Design your workload so that it provides the information necessary across all components (for example, metrics, logs, and traces) for you to understand its internal state. This enables you to provide effective responses when appropriate.

Best Practices:

- **Implement application telemetry**: Instrument your application code to emit information about its internal state, status, and achievement of business outcomes. For example, queue depth, error messages, and response times. Use this information to determine when a response is required.
- **Implement and configure workload telemetry**: Design and configure your workload to emit information about its internal state and current status. For example, API call volume, HTTP status codes, and scaling events. Use this information to help determine when a response is required.
- **Implement user activity telemetry**: Instrument your application code to emit information about user activity, for example, click streams, or started, abandoned, and completed transactions. Use this information to help understand how the application is used, patterns of usage, and to determine when a response is required.
- **Implement dependency telemetry**: Design and configure your workload to emit information about the status (for example, reachability or response time) of resources it depends on. Examples of external dependencies can include, external databases, DNS, and network connectivity. Use this information to determine when a response is required.
- **Implement transaction traceability**: Implement your application code and configure your workload components to emit information about the flow of transactions across the workload. Use this information to determine when a response is required and to assist you in identifying the factors contributing to an issue.

OPS 5 How do you reduce defects, ease remediation, and improve flow into production?

Adopt approaches that improve flow of changes into production, that enable refactoring, fast feedback on quality, and bug fixing. These accelerate beneficial changes entering production, limit issues deployed, and enable rapid identification and remediation of issues introduced through deployment activities.

Best Practices:

- **Use version control**: Use version control to enable tracking of changes and releases.
- **Test and validate changes**: Test and validate changes to help limit and detect errors. Automate testing to reduce errors caused by manual processes, and reduce the level of effort to test.
- **Use configuration management systems**: Use configuration management systems to make and track configuration changes. These systems reduce errors caused by manual processes and reduce the level of effort to deploy changes.
- **Use build and deployment management systems**: Use build and deployment management systems. These systems reduce errors caused by manual processes and reduce the level of effort to deploy changes.
- **Perform patch management**: Perform patch management to gain features, address issues, and remain compliant with governance. Automate patch management to reduce errors caused by manual processes, and reduce the level of effort to patch.
- **Share design standards**: Share best practices across teams to increase awareness and maximize the benefits of development efforts.
- **Implement practices to improve code quality**: Implement practices to improve code quality and minimize defects. For example, test-driven development, code reviews, and standards adoption.
- **Use multiple environments**: Use multiple environments to experiment, develop, and test your workload. Use increasing levels of controls as environments approach production to gain confidence your workload will operate as intended when deployed.
- **Make frequent, small, reversible changes**: Frequent, small, and reversible changes reduce the scope and impact of a change. This eases troubleshooting, enables faster remediation, and provides the option to roll back a change.

- **Fully automate integration and deployment**: Automate build, deployment, and testing of the workload. This reduces errors caused by manual processes and reduces the effort to deploy changes.

OPS 6 How do you mitigate deployment risks?

Adopt approaches that provide fast feedback on quality and enable rapid recovery from changes that do not have desired outcomes. Using these practices mitigates the impact of issues introduced through the deployment of changes.

Best Practices:

- **Plan for unsuccessful changes**: Plan to revert to a known good state, or remediate in the production environment if a change does not have the desired outcome. This preparation reduces recovery time through faster responses.
- **Test and validate changes**: Test changes and validate the results at all lifecycle stages to confirm new features and minimize the risk and impact of failed deployments.
- **Use deployment management systems**: Use deployment management systems to track and implement change. This reduces errors cause by manual processes and reduces the effort to deploy changes.
- **Test using limited deployments**: Test with limited deployments alongside existing systems to confirm desired outcomes prior to full scale deployment. For example, use deployment canary testing or one-box deployments.
- **Deploy using parallel environments**: Implement changes onto parallel environments, and then transition over to the new environment. Maintain the prior environment until there is confirmation of successful deployment. Doing so minimizes recovery time by enabling rollback to the previous environment.
- **Deploy frequent, small, reversible changes**: Use frequent, small, and reversible changes to reduce the scope of a change. This results in easier troubleshooting and faster remediation with the option to roll back a change.
- **Fully automate integration and deployment**: Automate build, deployment, and testing of the workload. This reduces errors cause by manual processes and reduces the effort to deploy changes.
- **Automate testing and rollback**: Automate testing of deployed environments to confirm desired outcomes. Automate rollback to previous known good state when outcomes are not achieved to minimize recovery time and reduce errors caused by manual processes.

OPS 7 How do you know that you are ready to support a workload?

Evaluate the operational readiness of your workload, processes and procedures, and personnel to understand the operational risks related to your workload.

Best Practices:

- **Ensure personnel capability**: Have a mechanism to validate that you have the appropriate number of trained personnel to provide support for operational needs. Train personnel and adjust personnel capacity as necessary to maintain effective support.
- **Ensure consistent review of operational readiness**: Ensure you have a consistent review of your readiness to operate a workload. Reviews must include, at a minimum, the operational readiness of the teams and the workload, and security requirements. Implement review activities in code and trigger automated review in response to events where appropriate, to ensure consistency, speed of execution, and reduce errors caused by manual processes.

- **Use runbooks to perform procedures**: Runbooks are documented procedures to achieve specific outcomes. Enable consistent and prompt responses to well-understood events by documenting procedures in runbooks. Implement runbooks as code and trigger the execution of runbooks in response to events where appropriate, to ensure consistency, speed responses, and reduce errors caused by manual processes.
- **Use playbooks to investigate issues**: Enable consistent and prompt responses to issues that are not well understood, by documenting the investigation process in playbooks. Playbooks are the predefined steps performed to identify the factors contributing to a failure scenario. The results from any process step are used to determine the next steps to take until the issue is identified or escalated.
- **Make informed decisions to deploy systems and changes**: Evaluate the capabilities of the team to support the workload and the workload's compliance with governance. Evaluate these against the benefits of deployment when determining whether to transition a system or change into production. Understand the benefits and risks to make informed decisions.

Operate

OPS 8 How do you understand the health of your workload?

Define, capture, and analyze workload metrics to gain visibility to workload events so that you can take appropriate action.

Best Practices:

- **Identify key performance indicators**: Identify key performance indicators (KPIs) based on desired business outcomes (for example, order rate, customer retention rate, and profit versus operating expense) and customer outcomes (for example, customer satisfaction). Evaluate KPIs to determine workload success.
- **Define workload metrics**: Define workload metrics to measure the achievement of KPIs (for example, abandoned shopping carts, orders placed, cost, price, and allocated workload expense). Define workload metrics to measure the health of the workload (for example, interface response time, error rate, requests made, requests completed, and utilization). Evaluate metrics to determine if the workload is achieving desired outcomes, and to understand the health of the workload.
- **Collect and analyze workload metrics**: Perform regular proactive reviews of metrics to identify trends and determine where appropriate responses are needed.
- **Establish workload metrics baselines**: Establish baselines for metrics to provide expected values as the basis for comparison and identification of under and over performing components. Identify thresholds for improvement, investigation, and intervention.
- **Learn expected patterns of activity for workload**: Establish patterns of workload activity to identify anomalous behavior so that you can respond appropriately if required.
- **Alert when workload outcomes are at risk**: Raise an alert when workload outcomes are at risk so that you can respond appropriately if necessary.
- **Alert when workload anomalies are detected**: Raise an alert when workload anomalies are detected so that you can respond appropriately if necessary.
- **Validate the achievement of outcomes and the effectiveness of KPIs and metrics** : Create a business-level view of your workload operations to help you determine if you are satisfying needs and to identify areas that need improvement to reach business goals. Validate the effectiveness of KPIs and metrics and revise them if necessary.

OPS 9 How do you understand the health of your operations?

Define, capture, and analyze operations metrics to gain visibility to operations events so that you can take appropriate action.

Best Practices:

- **Identify key performance indicators**: Identify key performance indicators (KPIs) based on desired business (for example, new features delivered) and customer outcomes (for example, customer support cases). Evaluate KPIs to determine operations success.
- **Define operations metrics**: Define operations metrics to measure the achievement of KPIs (for example, successful deployments, and failed deployments). Define operations metrics to measure the health of operations activities (for example, mean time to detect an incident (MTTD), and mean time to recovery (MTTR) from an incident). Evaluate metrics to determine if operations are achieving desired outcomes, and to understand the health of your operations activities.
- **Collect and analyze operations metrics**: Perform regular, proactive reviews of metrics to identify trends and determine where appropriate responses are needed.
- **Establish operations metrics baselines**: Establish baselines for metrics to provide expected values as the basis for comparison and identification of under and over performing operations activities.
- **Learn the expected patterns of activity for operations**: Establish patterns of operations activities to identify anomalous activity so that you can respond appropriately if necessary.
- **Alert when operations outcomes are at risk**: Raise an alert when operations outcomes are at risk so that you can respond appropriately if necessary.
- **Alert when operations anomalies are detected**: Raise an alert when operations anomalies are detected so that you can respond appropriately if necessary.
- **Validate the achievement of outcomes and the effectiveness of KPIs and metrics** : Create a business-level view of your operations activities to help you determine if you are satisfying needs and to identify areas that need improvement to reach business goals. Validate the effectiveness of KPIs and metrics and revise them if necessary.

OPS 10 How do you manage workload and operations events?

Prepare and validate procedures for responding to events to minimize their disruption to your workload.

Best Practices:

- **Use processes for event, incident, and problem management**: Have processes to address observed events, events that require intervention (incidents), and events that require intervention and either recur or cannot currently be resolved (problems). Use these processes to mitigate the impact of these events on the business and your customers by ensuring timely and appropriate responses.
- **Have a process per alert**: Have a well-defined response (runbook or playbook), with a specifically identified owner, for any event for which you raise an alert. This ensures effective and prompt responses to operations events and prevents actionable events from being obscured by less valuable notifications.
- **Prioritize operational events based on business impact**: Ensure that when multiple events require intervention, those that are most significant to the business are addressed first. For example, impacts can include loss of life or injury, financial loss, or damage to reputation or trust.
- **Define escalation paths**: Define escalation paths in your runbooks and playbooks, including what triggers escalation, and procedures for escalation. Specifically identify owners for each action to ensure effective and prompt responses to operations events.

- **Enable push notifications**: Communicate directly with your users (for example, with email or SMS) when the services they use are impacted, and again when the services return to normal operating conditions, to enable users to take appropriate action.
- **Communicate status through dashboards**: Provide dashboards tailored to their target audiences (for example, internal technical teams, leadership, and customers) to communicate the current operating status of the business and provide metrics of interest.
- **Automate responses to events**: Automate responses to events to reduce errors caused by manual processes, and to ensure prompt and consistent responses.

Evolve

OPS 11 How do you evolve operations?

Dedicate time and resources for continuous incremental improvement to evolve the effectiveness and efficiency of your operations.

Best Practices:

- **Have a process for continuous improvement**: Regularly evaluate and prioritize opportunities for improvement to focus efforts where they can provide the greatest benefits.
- **Perform post-incident analysis**: Review customer-impacting events, and identify the contributing factors and preventative actions. Use this information to develop mitigations to limit or prevent recurrence. Develop procedures for prompt and effective responses. Communicate contributing factors and corrective actions as appropriate, tailored to target audiences.
- **Implement feedback loops**: Include feedback loops in your procedures and workloads to help you identify issues and areas that need improvement.
- **Perform Knowledge Management**: Mechanisms exist for your team members to discover the information that they are looking for in a timely manner, access it, and identify that it's current and complete. Mechanisms are present to identify needed content, content in need of refresh, and content that should be archived so that it's no longer referenced.
- **Define drivers for improvement**: Identify drivers for improvement to help you evaluate and prioritize opportunities.
- **Validate insights**: Review your analysis results and responses with cross-functional teams and business owners. Use these reviews to establish common understanding, identify additional impacts, and determine courses of action. Adjust responses as appropriate.
- **Perform operations metrics reviews**: Regularly perform retrospective analysis of operations metrics with cross-team participants from different areas of the business. Use these reviews to identify opportunities for improvement, potential courses of action, and to share lessons learned.
- **Document and share lessons learned**: Document and share lessons learned from the execution of operations activities so that you can use them internally and across teams.
- **Allocate time to make improvements**: Dedicate time and resources within your processes to make continuous incremental improvements possible.

Security

Topics

Security Foundations

SEC 1 How do you securely operate your workload?

To operate your workload securely, you must apply overarching best practices to every area of security. Take requirements and processes that you have defined in operational excellence at an organizational and workload level, and apply them to all areas. Staying up to date with AWS and industry recommendations and threat intelligence helps you evolve your threat model and control objectives. Automating security processes, testing, and validation allow you to scale your security operations.

Best Practices:

- **Separate workloads using accounts**: Organize workloads in separate accounts and group accounts based on function or a common set of controls rather -than mirroring your company's reporting structure. Start with security and infrastructure in mind to enable your organization to set common guardrails as your workloads grow.
- **Secure AWS account**: Secure access to your accounts, for example by enabling MFA and restrict use of the root user, and configure account contacts.
- **Identify and validate control objectives**: Based on your compliance requirements and risks identified from your threat model, derive and validate the control objectives and controls that you need to apply to your workload. Ongoing validation of control objectives and controls help you measure the effectiveness of risk mitigation.
- **Keep up to date with security threats**: Recognize attack vectors by staying up to date with the latest security threats to help you define and implement appropriate controls.
- **Keep up to date with security recommendations**: Stay up to date with both AWS and industry security recommendations to evolve the security posture of your workload.
- **Automate testing and validation of security controls in pipelines**: Establish secure baselines and templates for security mechanisms that are tested and validated as part of your build, pipelines, and processes. Use tools and automation to test and validate all security controls continuously. For example, scan items such as machine images and infrastructure as code templates for security vulnerabilities, irregularities, and drift from an established baseline at each stage.
- **Identify and prioritize risks using a threat model**: Use a threat model to identify and maintain an up-to-date register of potential threats. Prioritize your threats and adapt your security controls to prevent, detect, and respond. Revisit and maintain this in the context of the evolving security landscape.
- **Evaluate and implement new security services and features regularly**: AWS and APN Partners constantly release new features and services that allow you to evolve the security posture of your workload.

Identity and Access Management

SEC 2 How do you manage authentication for people and machines?

There are two types of identities you need to manage when approaching operating secure AWS workloads. Understanding the type of identity you need to manage and grant access helps you ensure the right identities have access to the right resources under the right conditions.

Human Identities: Your administrators, developers, operators, and end users require an identity to access your AWS environments and applications. These are members of your organization, or external users with whom you collaborate, and who interact with your AWS resources via a web browser, client application, or interactive command-line tools.

Machine Identities: Your service applications, operational tools, and workloads require an identity to make requests to AWS services for example, to read data. These identities include machines running in your AWS environment such as Amazon EC2 instances or AWS Lambda functions. You may also manage machine identities for external parties who need access. Additionally, you may also have machines outside of AWS that need access to your AWS environment.

Best Practices:

- **Use strong sign-in mechanisms**: Enforce minimum password length, and educate users to avoid common or re-used passwords. Enforce multi-factor authentication (MFA) with software or hardware mechanisms to provide an additional layer.
- **Use temporary credentials**: Require identities to dynamically acquire temporary credentials. For workforce identities, use AWS Single Sign-On, or federation with IAM roles to access AWS accounts. For machine identities, require the use of IAM roles instead of long term access keys.
- **Store and use secrets securely**: For workforce and machine identities that require secrets such as passwords to third party applications, store them with automatic rotation using the latest industry standards in a specialized service.
- **Rely on a centralized identity provider**: For workforce identities, rely on an identity provider that enables you to manage identities in a centralized place. This enables you to create, manage, and revoke access from a single location making it easier to manage access. This reduces the requirement for multiple credentials and provides an opportunity to integrate with HR processes.
- **Audit and rotate credentials periodically**: When you cannot rely on temporary credentials and require long term credentials, audit credentials to ensure that the defined controls (for example, MFA) are enforced, rotated regularly, and have appropriate access level.
- **Leverage user groups and attributes**: Place users with common security requirements in groups defined by your identity provider, and put mechanisms in place to ensure that user attributes that may be used for access control (e.g., department or location) are correct and updated. Use these groups and attributes, rather than individual users, to control access. This allows you to manage access centrally by changing a user's group membership or attributes once, rather than updating many individual policies when a user's access needs change.

SEC 3 How do you manage permissions for people and machines?

Manage permissions to control access to people and machine identities that require access to AWS and your workload. Permissions control who can access what, and under what conditions.

Best Practices:

- **Define access requirements**: Each component or resource of your workload needs to be accessed by administrators, end users, or other components. Have a clear definition of who or what should have access to each component, choose the appropriate identity type and method of authentication and authorization.
- **Grant least privilege access**: Grant only the access that identities require by allowing access to specific actions on specific AWS resources under specific conditions. Rely on groups and identity attributes to dynamically set permissions at scale, rather than defining permissions for individual users. For example, you can allow a group of developers access to manage only resources for their project. This way, when a developer is removed from the group, access for the developer is revoked everywhere that group was used for access control, without requiring any changes to the access policies.
- **Establish emergency access process**: A process that allows emergency access to your workload in the unlikely event of an automated process or pipeline issue. This will help you rely on least privilege access, but ensure users can obtain the right level of access when they require it. For example, establish a process for administrators to verify and approve their request.
- **Reduce permissions continuously**: As teams and workloads determine what access they need, remove permissions they no longer use and establish review processes to achieve least privilege permissions. Continuously monitor and reduce unused identities and permissions.
- **Define permission guardrails for your organization**: Establish common controls that restrict access to all identities in your organization. For example, you can restrict access to specific AWS Regions, or prevent your operators from deleting common resources, such as an IAM role used for your central security team.
- **Manage access based on life cycle**: Integrate access controls with operator and application life cycle and your centralized federation provider. For example, remove a user's access when they leave the organization or change roles.
- **Analyze public and cross account access**: Continuously monitor findings that highlight public and cross account access. Reduce public access and cross account access to only resources that require this type of access.
- **Share resources securely**: Govern the consumption of shared resources across accounts or within your AWS Organization. Monitor shared resources and review shared resource access.

Detection

SEC 4 How do you detect and investigate security events?

Capture and analyze events from logs and metrics to gain visibility. Take action on security events and potential threats to help secure your workload.

Best Practices:

- **Configure service and application logging**: Configure logging throughout the workload, including application logs, resource logs, and AWS service logs. For example, ensure that AWS CloudTrail, Amazon CloudWatch Logs, Amazon GuardDuty and AWS Security Hub are enabled for all accounts within your organization.
- **Analyze logs, findings, and metrics centrally**: All logs, metrics, and telemetry should be collected centrally, and automatically analyzed to detect anomalies and indicators of unauthorized activity. A dashboard can provide you easy to access insight into real-time health. For example, ensure that Amazon GuardDuty and Security Hub logs are sent to a central location for alerting and analysis.
- **Automate response to events**: Using automation to investigate and remediate events reduces human effort and error, and enables you to scale investigation capabilities. Regular reviews will help you tune automation tools, and continuously iterate. For example, automate responses to Amazon GuardDuty events by automating the first investigation step, then iterate to gradually remove human effort.

- **Implement actionable security events**: Create alerts that are sent to and can be actioned by your team. Ensure that alerts include relevant information for the team to take action. For example, ensure that Amazon GuardDuty and AWS Security Hub alerts are sent to the team to action, or sent to response automation tooling with the team remaining informed by messaging from the automation framework.

Infrastructure Protection

SEC 5 How do you protect your network resources?

Any workload that has some form of network connectivity, whether it's the internet or a private network, requires multiple layers of defense to help protect from external and internal network-based threats.

Best Practices:

- **Create network layers**: Group components that share reachability requirements into layers. For example, a database cluster in a VPC with no need for internet access should be placed in subnets with no route to or from the internet. In a serverless workload operating without a VPC, similar layering and segmentation with microservices can achieve the same goal.
- **Control traffic at all layers**: Apply controls with a defense in depth approach for both inbound and outbound traffic. For example, for Amazon Virtual Private Cloud (VPC) this includes security groups, Network ACLs, and subnets. For AWS Lambda, consider running in your private VPC with VPC-based controls.
- **Automate network protection**: Automate protection mechanisms to provide a self-defending network based on threat intelligence and anomaly detection. For example, intrusion detection and prevention tools that can pro-actively adapt to current threats and reduce their impact.
- **Implement inspection and protection**: Inspect and filter your traffic at each layer. For example, use a web application firewall to help protect against inadvertent access at the application network layer. For Lambda functions, third-party tools can add application-layer firewalling to your runtime environment.

SEC 6 How do you protect your compute resources?

Compute resources in your workload require multiple layers of defense to help protect from external and internal threats. Compute resources include EC2 instances, containers, AWS Lambda functions, database services, IoT devices, and more.

Best Practices:

- **Perform vulnerability management**: Frequently scan and patch for vulnerabilities in your code, dependencies, and in your infrastructure to help protect against new threats.
- **Reduce attack surface**: Reduce your attack surface by hardening operating systems, minimizing components, libraries, and externally consumable services in use.
- **Implement managed services**: Implement services that manage resources, such as Amazon RDS, AWS Lambda, and Amazon ECS, to reduce your security maintenance tasks as part of the shared responsibility model.
- **Automate compute protection**: Automate your protective compute mechanisms including vulnerability management, reduction in attack surface, and management of resources.

- **Enable people to perform actions at a distance**: Removing the ability for interactive access reduces the risk of human error, and the potential for manual configuration or management. For example, use a change management workflow to deploy EC2 instances using infrastructure as code, then manage EC2 instances using tools instead of allowing direct access or a bastion host.
- **Validate software integrity**: Implement mechanisms (for example, code signing) to validate that the software, code, and libraries used in the workload are from trusted sources and have not been tampered with.

Data Protection

SEC 7 How do you classify your data?

Classification provides a way to categorize data, based on criticality and sensitivity in order to help you determine appropriate protection and retention controls.

Best Practices:

- **Identify the data within your workload**: This includes the type and classification of data, the associated business processes. data owner, applicable legal and compliance requirements, where it's stored, and the resulting controls that are needed to be enforced. This may include classifications to indicate if the data is intended to be publicly available, if the data is internal use only such as customer personally identifiable information (PII), or if the data is for more restricted access such as intellectual property, legally privileged or marked sensitive, and more.
- **Define data protection controls**: Protect data according to its classification level. For example, secure data classified as public by using relevant recommendations while protecting sensitive data with additional controls.
- **Automate identification and classification**: Automate identification and classification of data to reduce the risk of human error from manual interactions.
- **Define data lifecycle management**: Your defined lifecycle strategy should be based on sensitivity level, as well as legal and organization requirements. Aspects including the duration you retain data for, data destruction, data access management, data transformation, and data sharing should be considered.

SEC 8 How do you protect your data at rest?

Protect your data at rest by implementing multiple controls, to reduce the risk of unauthorized access or mishandling.

Best Practices:

- **Implement secure key management**: Encryption keys must be stored securely, with strict access control, for example, by using a key management service such as AWS KMS. Consider using different keys, and access control to the keys, combined with the AWS IAM and resource policies, to align with data classification levels and segregation requirements.
- **Enforce encryption at rest**: Enforce your encryption requirements based on the latest standards and recommendations to help protect your data at rest.
- **Automate data at rest protection**: Use automated tools to validate and enforce data at rest protection continuously, for example, verify that there are only encrypted storage resources.

- **Enforce access control**: Enforce access control with least privileges and mechanisms, including backups, isolation, and versioning, to help protect your data at rest. Consider what data you have that is publicly accessible.
- **Use mechanisms to keep people away from data**: Keep all users away from directly accessing sensitive data and systems under normal operational circumstances. For example, provide a dashboard instead of direct access to a data store to run queries. Where CI/CD pipelines are not used, determine which controls and processes are required to adequately provide a normally disabled break-glass access mechanism.

SEC 9 How do you protect your data in transit?

Protect your data in transit by implementing multiple controls to reduce the risk of unauthorized access or loss.

Best Practices:

- **Implement secure key and certificate management**: Store encryption keys and certificates securely and rotate them at appropriate time intervals while applying strict access control; for example, by using a certificate management service, such as AWS Certificate Manager (ACM).
- **Enforce encryption in transit**: Enforce your defined encryption requirements based on appropriate standards and recommendations to help you meet your organizational, legal, and compliance requirements.
- **Automate detection of unintended data access**: Use tools such as GuardDuty to automatically detect attempts to move data outside of defined boundaries based on data classification level, for example, to detect a trojan that is copying data to an unknown or untrusted network using the DNS protocol.
- **Authenticate network communications**: Verify the identity of communications by using protocols that support authentication, such as Transport Layer Security (TLS) or IPsec.

Incident Response

SEC 10 How do you anticipate, respond to, and recover from incidents?

Preparation is critical to timely and effective investigation, response to, and recovery from security incidents to help minimize disruption to your organization.

Best Practices:

- **Identify key personnel and external resources**: Identify internal and external personnel, resources, and legal obligations that would help your organization respond to an incident.
- **Develop incident management plans**: Create plans to help you respond to, communicate during, and recover from an incident. For example, you can start an incident response plan with the most likely scenarios for your workload and organization. Include how you would communicate and escalate both internally and externally.
- **Prepare forensic capabilities**: Identify and prepare forensic investigation capabilities that are suitable, including external specialists, tools, and automation.
- **Automate containment capability**: Automate containment and recovery of an incident to reduce response times and organizational impact.
- **Pre-provision access**: Ensure that incident responders have the correct access pre-provisioned into AWS to reduce the time for investigation through to recovery.

- **Pre-deploy tools**: Ensure that security personnel have the right tools pre-deployed into AWS to reduce the time for investigation through to recovery.
- **Run game days**: Practice incident response game days (simulations) regularly, incorporate lessons learned into your incident management plans, and continuously improve.

Reliability

Topics

Foundations

REL 1 How do you manage service quotas and constraints?

For cloud-based workload architectures, there are service quotas (which are also referred to as service limits). These quotas exist to prevent accidentally provisioning more resources than you need and to limit request rates on API operations so as to protect services from abuse. There are also resource constraints, for example, the rate that you can push bits down a fiber-optic cable, or the amount of storage on a physical disk.

Best Practices:

- **Aware of service quotas and constraints**: You are aware of your default quotas and quota increase requests for your workload architecture. You additionally know which resource constraints, such as disk or network, are potentially impactful.
- **Manage service quotas across accounts and regions**: If you are using multiple AWS accounts or AWS Regions, ensure that you request the appropriate quotas in all environments in which your production workloads run.
- **Accommodate fixed service quotas and constraints through architecture**: Be aware of unchangeable service quotas and physical resources, and architect to prevent these from impacting reliability.
- **Monitor and manage quotas**: Evaluate your potential usage and increase your quotas appropriately allowing for planned growth in usage.
- **Automate quota management**: Implement tools to alert you when thresholds are being approached. By using AWS Service Quotas APIs, you can automate quota increase requests.
- **Ensure that a sufficient gap exists between the current quotas and the maximum usage to accommodate failover**: When a resource fails, it may still be counted against quotas until its successfully terminated. Ensure that your quotas cover the overlap of all failed resources with replacements before the failed resources are terminated. You should consider an Availability Zone failure when calculating this gap.

REL 2 How do you plan your network topology?

Workloads often exist in multiple environments. These include multiple cloud environments (both publicly accessible and private) and possibly your existing data center infrastructure. Plans must include network considerations such as intra- and inter-system connectivity, public IP address management, private IP address management, and domain name resolution.

Best Practices:

- **Use highly available network connectivity for your workload public endpoints**: These endpoints and the routing to them must be highly available. To achieve this, use highly available DNS, content delivery networks (CDNs), API Gateway, load balancing, or reverse proxies.
- **Provision redundant connectivity between private networks in the cloud and on-premises environments**: Use multiple AWS Direct Connect (DX) connections or VPN tunnels between separately deployed private networks. Use multiple DX locations for high availability. If using multiple AWS Regions, ensure redundancy in at least two of them. You might want to evaluate AWS Marketplace appliances that terminate VPNs. If you use AWS Marketplace appliances, deploy redundant instances for high availability in different Availability Zones.
- **Ensure IP subnet allocation accounts for expansion and availability**: Amazon VPC IP address ranges must be large enough to accommodate workload requirements, including factoring in future expansion and allocation of IP addresses to subnets across Availability Zones. This includes load balancers, EC2 instances, and container-based applications.
- **Prefer hub-and-spoke topologies over many-to-many mesh**: If more than two network address spaces (for example, VPCs and on-premises networks) are connected via VPC peering, AWS Direct Connect, or VPN, then use a hub-and-spoke model, like that provided by AWS Transit Gateway.
- **Enforce non-overlapping private IP address ranges in all private address spaces where they are connected**: The IP address ranges of each of your VPCs must not overlap when peered or connected via VPN. You must similarly avoid IP address conflicts between a VPC and on-premises environments or with other cloud providers that you use. You must also have a way to allocate private IP address ranges when needed.

Workload Architecture

REL 3 How do you design your workload service architecture?

Build highly scalable and reliable workloads using a service-oriented architecture (SOA) or a microservices architecture. Service-oriented architecture (SOA) is the practice of making software components reusable via service interfaces. Microservices architecture goes further to make components smaller and simpler.

Best Practices:

- **Choose how to segment your workload**: Monolithic architecture should be avoided. Instead, you should choose between SOA and microservices. When making each choice, balance the benefits against the complexities—what is right for a new product racing to first launch is different than what a workload built to scale from the start needs. The benefits of using smaller segments include greater agility, organizational flexibility, and scalability. Complexities include possible increased latency, more complex debugging, and increased operational burden
- **Build services focused on specific business domains and functionality**: SOA builds services with well-delineated functions defined by business needs. Microservices use domain models and bounded context to limit this further so that each service does just one thing. Focusing on specific functionality enables you to differentiate the reliability requirements of different services, and target investments more specifically. A concise business problem and having a small team associated with each service also enables easier organizational scaling.
- **Provide service contracts per API**: Service contracts are documented agreements between teams on service integration and include a machine-readable API definition, rate limits, and performance expectations. A versioning strategy allows clients to continue using the existing API and migrate their applications to the newer API when they are ready. Deployment can happen anytime, as long as the contract is not violated. The service provider team can use the technology stack of their choice to satisfy the API contract. Similarly, the service consumer can use their own technology.

REL 4 How do you design interactions in a distributed system to prevent failures?

Distributed systems rely on communications networks to interconnect components, such as servers or services. Your workload must operate reliably despite data loss or latency in these networks. Components of the distributed system must operate in a way that does not negatively impact other components or the workload. These best practices prevent failures and improve mean time between failures (MTBF).

Best Practices:

- **Identify which kind of distributed system is required**: Hard real-time distributed systems require responses to be given synchronously and rapidly, while soft real-time systems have a more generous time window of minutes or more for response. Offline systems handle responses through batch or asynchronous processing. Hard real-time distributed systems have the most stringent reliability requirements.
- **Implement loosely coupled dependencies**: Dependencies such as queuing systems, streaming systems, workflows, and load balancers are loosely coupled. Loose coupling helps isolate behavior of a component from other components that depend on it, increasing resiliency and agility
- **Make all responses idempotent**: An idempotent service promises that each request is completed exactly once, such that making multiple identical requests has the same effect as making a single request. An idempotent service makes it easier for a client to implement retries without fear that a request will be erroneously processed multiple times. To do this, clients can issue API requests with an idempotency token—the same token is used whenever the request is repeated. An idempotent service API uses the token to return a response identical to the response that was returned the first time that the request was completed.
- **Do constant work**: Systems can fail when there are large, rapid changes in load. For example, if your workload is doing a health check that monitors the health of thousands of servers, it should send the same size payload (a full snapshot of the current state) each time. Whether no servers are failing, or all of them, the health check system is doing constant work with no large, rapid changes.

REL 5 How do you design interactions in a distributed system to mitigate or withstand failures?

Distributed systems rely on communications networks to interconnect components (such as servers or services). Your workload must operate reliably despite data loss or latency over these networks. Components of the distributed system must operate in a way that does not negatively impact other components or the workload. These best practices enable workloads to withstand stresses or failures, more quickly recover from them, and mitigate the impact of such impairments. The result is improved mean time to recovery (MTTR).

Best Practices:

- **Implement graceful degradation to transform applicable hard dependencies into soft dependencies**: When a component's dependencies are unhealthy, the component itself can still function, although in a degraded manner. For example, when a dependency call fails, failover to a predetermined static response.
- **Throttle requests**: This is a mitigation pattern to respond to an unexpected increase in demand. Some requests are honored but those over a defined limit are rejected and return a message indicating they have been throttled. The expectation on clients is that they will back off and abandon the request or try again at a slower rate.
- **Control and limit retry calls**: Use exponential backoff to retry after progressively longer intervals. Introduce jitter to randomize those retry intervals, and limit the maximum number of retries.

- **Fail fast and limit queues**: If the workload is unable to respond successfully to a request, then fail fast. This allows the releasing of resources associated with a request, and permits the service to recover if it's running out of resources. If the workload is able to respond successfully but the rate of requests is too high, then use a queue to buffer requests instead. However, do not allow long queues that can result in serving stale requests that the client has already given up on.
- **Set client timeouts**: Set timeouts appropriately, verify them systematically, and do not rely on default values as they are generally set too high
- **Make services stateless where possible**: Services should either not require state, or should offload state such that between different client requests, there is no dependence on locally stored data on disk or in memory. This enables servers to be replaced at will without causing an availability impact. Amazon ElastiCache or Amazon DynamoDB are good destinations for offloaded state.
- **Implement emergency levers**: These are rapid processes that may mitigate availability impact on your workload. They can be operated in the absence of a root cause. An ideal emergency lever reduces the cognitive burden on the resolvers to zero by providing fully deterministic activation and deactivation criteria. Example levers include blocking all robot traffic or serving a static response. Levers are often manual, but they can also be automated.

Change Management

REL 6 How do you monitor workload resources?

Logs and metrics are powerful tools to gain insight into the health of your workload. You can configure your workload to monitor logs and metrics and send notifications when thresholds are crossed or significant events occur. Monitoring enables your workload to recognize when low-performance thresholds are crossed or failures occur, so it can recover automatically in response.

Best Practices:

- **Monitor all components for the workload (Generation)**: Monitor the components of the workload with Amazon CloudWatch or third-party tools. Monitor AWS services with AWS Personal Health Dashboard
- **Define and calculate metrics (Aggregation)**: Store log data and apply filters where necessary to calculate metrics, such as counts of a specific log event, or latency calculated from log event timestamps
- **Send notifications (Real-time processing and alarming)**: Organizations that need to know, receive notifications when significant events occur
- **Automate responses (Real-time processing and alarming)**: Use automation to take action when an event is detected, for example, to replace failed components
- **Storage and Analytics**: Collect log files and metrics histories and analyze these for broader trends and workload insights
- **Conduct reviews regularly**: Frequently review how workload monitoring is implemented and update it based on significant events and changes
- **Monitor end-to-end tracing of requests through your system**: Use AWS X-Ray or third-party tools so that developers can more easily analyze and debug distributed systems to understand how their applications and its underlying services are performing

REL 7 How do you design your workload to adapt to changes in demand?

A scalable workload provides elasticity to add or remove resources automatically so that they closely match the current demand at any given point in time.

Best Practices:

- **Use automation when obtaining or scaling resources**: When replacing impaired resources or scaling your workload, automate the process by using managed AWS services, such as Amazon S3 and AWS Auto Scaling. You can also use third-party tools and AWS SDKs to automate scaling.
- **Obtain resources upon detection of impairment to a workload**: Scale resources reactively when necessary if availability is impacted, to restore workload availability.
- **Obtain resources upon detection that more resources are needed for a workload**: Scale resources proactively to meet demand and avoid availability impact.
- **Load test your workload**: Adopt a load testing methodology to measure if scaling activity meets workload requirements.

REL 8 How do you implement change?

Controlled changes are necessary to deploy new functionality, and to ensure that the workloads and the operating environment are running known software and can be patched or replaced in a predictable manner. If these changes are uncontrolled, then it makes it difficult to predict the effect of these changes, or to address issues that arise because of them.

Best Practices:

- **Use runbooks for standard activities such as deployment**: Runbooks are the predefined steps used to achieve specific outcomes. Use runbooks to perform standard activities, whether done manually or automatically. Examples include deploying a workload, patching it, or making DNS modifications.
- **Integrate functional testing as part of your deployment**: Functional tests are run as part of automated deployment. If success criteria are not met, the pipeline is halted or rolled back.
- **Integrate resiliency testing as part of your deployment**: Resiliency tests (as part of chaos engineering) are run as part of the automated deployment pipeline in a pre-prod environment.
- **Deploy using immutable infrastructure**: This is a model that mandates that no updates, security patches, or configuration changes happen in-place on production workloads. When a change is needed, the architecture is built onto new infrastructure and deployed into production.
- **Deploy changes with automation**: Deployments and patching are automated to eliminate negative impact.

Failure Management

REL 9 How do you back up data?

Back up data, applications, and configuration to meet your requirements for recovery time objectives (RTO) and recovery point objectives (RPO).

Best Practices:

- **Identify and back up all data that needs to be backed up, or reproduce the data from sources**: Amazon S3 can be used as a backup destination for multiple data sources. AWS services such as Amazon EBS, Amazon RDS, and Amazon DynamoDB have built in capabilities to create backups. Third-party backup software can also be used. Alternatively, if the data can be reproduced from other sources to meet RPO, you might not require a backup
- **Secure and encrypt backups**: Detect access using authentication and authorization, such as AWS IAM, and detect data integrity compromise by using encryption.

- **Perform data backup automatically**: Configure backups to be taken automatically based on a periodic schedule, or by changes in the dataset. RDS instances, EBS volumes, DynamoDB tables, and S3 objects can all be configured for automatic backup. AWS Marketplace solutions or third-party solutions can also be used.
- **Perform periodic recovery of the data to verify backup integrity and processes**: Validate that your backup process implementation meets your recovery time objectives (RTO) and recovery point objectives (RPO) by performing a recovery test.

REL 10 How do you use fault isolation to protect your workload?

Fault isolated boundaries limit the effect of a failure within a workload to a limited number of components. Components outside of the boundary are unaffected by the failure. Using multiple fault isolated boundaries, you can limit the impact on your workload.

Best Practices:

- **Deploy the workload to multiple locations**: Distribute workload data and resources across multiple Availability Zones or, where necessary, across AWS Regions. These locations can be as diverse as required.
- **Automate recovery for components constrained to a single location**: If components of the workload can only run in a single Availability Zone or on-premises data center, you must implement the capability to do a complete rebuild of the workload within your defined recovery objectives.
- **Use bulkhead architectures to limit scope of impact**: Like the bulkheads on a ship, this pattern ensures that a failure is contained to a small subset of requests/users so that the number of impaired requests is limited, and most can continue without error. Bulkheads for data are often called partitions, while bulkheads for services are known as cells.

REL 11 How do you design your workload to withstand component failures?

Workloads with a requirement for high availability and low mean time to recovery (MTTR) must be architected for resiliency.

Best Practices:

- **Monitor all components of the workload to detect failures**: Continuously monitor the health of your workload so that you and your automated systems are aware of degradation or complete failure as soon as they occur. Monitor for key performance indicators (KPIs) based on business value.
- **Fail over to healthy resources**: Ensure that if a resource failure occurs, that healthy resources can continue to serve requests. For location failures (such as Availability Zone or AWS Region) ensure you have systems in place to failover to healthy resources in unimpaired locations.
- **Automate healing on all layers**: Upon detection of a failure, use automated capabilities to perform actions to remediate.
- **Use static stability to prevent bimodal behavior**: Bimodal behavior is when your workload exhibits different behavior under normal and failure modes, for example, relying on launching new instances if an Availability Zone fails. You should instead build workloads that are statically stable and operate in only one mode. In this case, provision enough instances in each Availability Zone to handle the workload load if one AZ were removed and then use Elastic Load Balancing or Amazon Route 53 health checks to shift load away from the impaired instances.
- **Send notifications when events impact availability**: Notifications are sent upon the detection of significant events, even if the issue caused by the event was automatically resolved.

REL 12 How do you test reliability?

After you have designed your workload to be resilient to the stresses of production, testing is the only way to ensure that it will operate as designed, and deliver the resiliency you expect.

Best Practices:

- **Use playbooks to investigate failures**: Enable consistent and prompt responses to failure scenarios that are not well understood, by documenting the investigation process in playbooks. Playbooks are the predefined steps performed to identify the factors contributing to a failure scenario. The results from any process step are used to determine the next steps to take until the issue is identified or escalated.
- **Perform post-incident analysis**: Review customer-impacting events, and identify the contributing factors and preventative action items. Use this information to develop mitigations to limit or prevent recurrence. Develop procedures for prompt and effective responses. Communicate contributing factors and corrective actions as appropriate, tailored to target audiences. Have a method to communicate these causes to others as needed.
- **Test functional requirements**: These include unit tests and integration tests that validate required functionality.
- **Test scaling and performance requirements**: This includes load testing to validate that the workload meets scaling and performance requirements.
- **Test resiliency using chaos engineering**: Run tests that inject failures regularly into pre-production and production environments. Hypothesize how your workload will react to the failure, then compare your hypothesis to the testing results and iterate if they do not match. Ensure that production testing does not impact users.
- **Conduct game days regularly**: Use game days to regularly exercise your procedures for responding to events and failures as close to production as possible (including in production environments) with the people who will be involved in actual failure scenarios. Game days enforce measures to ensure that production testing does not impact users.

REL 13 How do you plan for disaster recovery (DR)?

Having backups and redundant workload components in place is the start of your DR strategy. RTO and RPO are your objectives for restoration of your workload. Set these based on business needs. Implement a strategy to meet these objectives, considering locations and function of workload resources and data. The probability of disruption and cost of recovery are also key factors that help to inform the business value of providing disaster recovery for a workload.

Best Practices:

- **Define recovery objectives for downtime and data loss**: The workload has a recovery time objective (RTO) and recovery point objective (RPO).
- **Use defined recovery strategies to meet the recovery objectives**: A disaster recovery (DR) strategy has been defined to meet objectives. Choose a strategy such as: backup and restore, active/passive (pilot light or warm standby), or active/active.
- **Test disaster recovery implementation to validate the implementation**: Regularly test failover to DR to ensure that RTO and RPO are met.
- **Manage configuration drift at the DR site or region**: Ensure that the infrastructure, data, and configuration are as needed at the DR site or region. For example, check that AMIs and service quotas are up to date.

- **Automate recovery**: Use AWS or third-party tools to automate system recovery and route traffic to the DR site or region.

Performance Efficiency

Topics

Selection

PERF 1 How do you select the best performing architecture?

Often, multiple approaches are required for optimal performance across a workload. Well-architected systems use multiple solutions and features to improve performance.

Best Practices:

- **Understand the available services and resources**: Learn about and understand the wide range of services and resources available in the cloud. Identify the relevant services and configuration options for your workload, and understand how to achieve optimal performance.
- **Define a process for architectural choices**: Use internal experience and knowledge of the cloud, or external resources such as published use cases, relevant documentation, or whitepapers to define a process to choose resources and services. You should define a process that encourages experimentation and benchmarking with the services that could be used in your workload.
- **Factor cost requirements into decisions** : Workloads often have cost requirements for operation. Use internal cost controls to select resource types and sizes based on predicted resource need.
- **Use policies or reference architectures**: Maximize performance and efficiency by evaluating internal policies and existing reference architectures and using your analysis to select services and configurations for your workload.
- **Use guidance from your cloud provider or an appropriate partner**: Use cloud company resources, such as solutions architects, professional services, or an appropriate partner to guide your decisions. These resources can help review and improve your architecture for optimal performance.
- **Benchmark existing workloads**: Benchmark the performance of an existing workload to understand how it performs on the cloud. Use the data collected from benchmarks to drive architectural decisions.
- **Load test your workload**: Deploy your latest workload architecture on the cloud using different resource types and sizes. Monitor the deployment to capture performance metrics that identify bottlenecks or excess capacity. Use this performance information to design or improve your architecture and resource selection.

PERF 2 How do you select your compute solution?

The optimal compute solution for a workload varies based on application design, usage patterns, and configuration settings. Architectures can use different compute solutions for various components and enable different features to improve performance. Selecting the wrong compute solution for an architecture can lead to lower performance efficiency.

Best Practices:

- **Evaluate the available compute options**: Understand the performance characteristics of the compute-related options available to you. Know how instances, containers, and functions work, and what advantages, or disadvantages, they bring to your workload.
- **Understand the available compute configuration options**: Understand how various options complement your workload, and which configuration options are best for your system. Examples of these options include instance family, sizes, features (GPU, I/O), function sizes, container instances, and single versus multi-tenancy.
- **Collect compute-related metrics**: One of the best ways to understand how your compute systems are performing is to record and track the true utilization of various resources. This data can be used to make more accurate determinations about resource requirements.
- **Determine the required configuration by right-sizing**: Analyze the various performance characteristics of your workload and how these characteristics relate to memory, network, and CPU usage. Use this data to choose resources that best match your workload's profile. For example, a memory-intensive workload, such as a database, could be served best by the r-family of instances. However, a bursting workload can benefit more from an elastic container system.
- **Use the available elasticity of resources**: The cloud provides the flexibility to expand or reduce your resources dynamically through a variety of mechanisms to meet changes in demand. Combined with compute-related metrics, a workload can automatically respond to changes and utilize the optimal set of resources to achieve its goal.
- **Re-evaluate compute needs based on metrics**: Use system-level metrics to identify the behavior and requirements of your workload over time. Evaluate your workload's needs by comparing the available resources with these requirements and make changes to your compute environment to best match your workload's profile. For example, over time a system might be observed to be more memory-intensive than initially thought, so moving to a different instance family or size could improve both performance and efficiency.

PERF 3 How do you select your storage solution?

The optimal storage solution for a system varies based on the kind of access method (block, file, or object), patterns of access (random or sequential), required throughput, frequency of access (online, offline, archival), frequency of update (WORM, dynamic), and availability and durability constraints. Well-architected systems use multiple storage solutions and enable different features to improve performance and use resources efficiently.

Best Practices:

- **Understand storage characteristics and requirements**: Understand the different characteristics (for example, shareable, file size, cache size, access patterns, latency, throughput, and persistence of data) that are required to select the services that best fit your workload, such as object storage, block storage, file storage, or instance storage.
- **Evaluate available configuration options**: Evaluate the various characteristics and configuration options and how they relate to storage. Understand where and how to use provisioned IOPS, SSDs, magnetic storage, object storage, archival storage, or ephemeral storage to optimize storage space and performance for your workload.
- **Make decisions based on access patterns and metrics**: Choose storage systems based on your workload's access patterns and configure them by determining how the workload accesses data. Increase storage efficiency by choosing object storage over block storage. Configure the storage options you choose to match your data access patterns.

PERF 4 How do you select your database solution?

The optimal database solution for a system varies based on requirements for availability, consistency, partition tolerance, latency, durability, scalability, and query capability. Many systems use different database solutions for various subsystems and enable different features to improve performance. Selecting the wrong database solution and features for a system can lead to lower performance efficiency.

Best Practices:

- **Understand data characteristics**: Understand the different characteristics of data in your workload. Determine if the workload requires transactions, how it interacts with data, and what its performance demands are. Use this data to select the best performing database approach for your workload (for example, relational databases, NoSQL Key-value, document, wide column, graph, time series, or in-memory storage).
- **Evaluate the available options**: Evaluate the services and storage options that are available as part of the selection process for your workload's storage mechanisms. Understand how, and when, to use a given service or system for data storage. Learn about available configuration options that can optimize database performance or efficiency, such as provisioned IOPs, memory and compute resources, and caching.
- **Collect and record database performance metrics**: Use tools, libraries, and systems that record performance measurements related to database performance. For example, measure transactions per second, slow queries, or system latency introduced when accessing the database. Use this data to understand the performance of your database systems.
- **Choose data storage based on access patterns**: Use the access patterns of the workload to decide which services and technologies to use. For example, utilize a relational database for workloads that require transactions, or a key-value store that provides higher throughput but is eventually consistent where applicable.
- **Optimize data storage based on access patterns and metrics**: Use performance characteristics and access patterns that optimize how data is stored or queried to achieve the best possible performance. Measure how optimizations such as indexing, key distribution, data warehouse design, or caching strategies impact system performance or overall efficiency.

PERF 5 How do you configure your networking solution?

The optimal network solution for a workload varies based on latency, throughput requirements, jitter, and bandwidth. Physical constraints, such as user or on-premises resources, determine location options. These constraints can be offset with edge locations or resource placement.

Best Practices:

- **Understand how networking impacts performance**: Analyze and understand how network-related decisions impact workload performance. For example, network latency often impacts the user experience, and using the wrong protocols can starve network capacity through excessive overhead.
- **Evaluate available networking features**: Evaluate networking features in the cloud that may increase performance. Measure the impact of these features through testing, metrics, and analysis. For example, take advantage of network-level features that are available to reduce latency, network distance, or jitter.
- **Choose appropriately sized dedicated connectivity or VPN for hybrid workloads**: When there is a requirement for on-premise communication, ensure that you have adequate bandwidth for workload performance. Based on bandwidth requirements, a single dedicated connection or a single VPN might not be enough, and you must enable traffic load balancing across multiple connections.

- **Leverage load-balancing and encryption offloading**: Distribute traffic across multiple resources or services to allow your workload to take advantage of the elasticity that the cloud provides. You can also use load balancing for offloading encryption termination to improve performance and to manage and route traffic effectively.
- **Choose network protocols to improve performance**: Make decisions about protocols for communication between systems and networks based on the impact to the workload's performance.
- **Choose your workload's location based on network requirements**: Use the cloud location options available to reduce network latency or improve throughput. Utilize AWS Regions, Availability Zones, placement groups, and edge locations such as Outposts, Local Zones, and Wavelength, to reduce network latency or improve throughput.
- **Optimize network configuration based on metrics**: Use collected and analyzed data to make informed decisions about optimizing your network configuration. Measure the impact of those changes and use the impact measurements to make future decisions.

Review

PERF 6 How do you evolve your workload to take advantage of new releases?

When architecting workloads, there are finite options that you can choose from. However, over time, new technologies and approaches become available that could improve the performance of your workload.

Best Practices:

- **Stay up-to-date on new resources and services**: Evaluate ways to improve performance as new services, design patterns, and product offerings become available. Determine which of these could improve performance or increase the efficiency of the workload through ad-hoc evaluation, internal discussion, or external analysis.
- **Define a process to improve workload performance**: Define a process to evaluate new services, design patterns, resource types, and configurations as they become available. For example, run existing performance tests on new instance offerings to determine their potential to improve your workload.
- **Evolve workload performance over time**: As an organization, use the information gathered through the evaluation process to actively drive adoption of new services or resources when they become available.

Monitoring

PERF 7 How do you monitor your resources to ensure they are performing?

System performance can degrade over time. Monitor system performance to identify degradation and remediate internal or external factors, such as the operating system or application load.

Best Practices:

- **Record performance-related metrics**: Use a monitoring and observability service to record performance-related metrics. For example, record database transactions, slow queries, I/O latency, HTTP request throughput, service latency, or other key data.
- **Analyze metrics when events or incidents occur**: In response to (or during) an event or incident, use monitoring dashboards or reports to understand and diagnose the impact. These views provide insight into which portions of the workload are not performing as expected.

- **Establish Key Performance Indicators (KPIs) to measure workload performance**: Identify the KPIs that indicate whether the workload is performing as intended. For example, an API-based workload might use overall response latency as an indication of overall performance, and an e-commerce site might choose to use the number of purchases as its KPI.
- **Use monitoring to generate alarm-based notifications**: Using the performance-related key performance indicators (KPIs) that you defined, use a monitoring system that generates alarms automatically when these measurements are outside expected boundaries.
- **Review metrics at regular intervals**: As routine maintenance, or in response to events or incidents, review which metrics are collected. Use these reviews to identify which metrics were key in addressing issues and which additional metrics, if they were being tracked, would help to identify, address, or prevent issues.
- **Monitor and alarm proactively**: Use key performance indicators (KPIs), combined with monitoring and alerting systems, to proactively address performance-related issues. Use alarms to trigger automated actions to remediate issues where possible. Escalate the alarm to those able to respond if automated response is not possible. For example, you may have a system that can predict expected key performance indicators (KPI) values and alarm when they breach certain thresholds, or a tool that can automatically halt or roll back deployments if KPIs are outside of expected values.

Tradeoffs

PERF 8 How do you use tradeoffs to improve performance?

When architecting solutions, determining tradeoffs enables you to select an optimal approach. Often you can improve performance by trading consistency, durability, and space for time and latency.

Best Practices:

- **Understand the areas where performance is most critical**: Understand and identify areas where increasing the performance of your workload will have a positive impact on efficiency or customer experience. For example, a website that has a large amount of customer interaction can benefit from using edge services to move content delivery closer to customers.
- **Learn about design patterns and services**: Research and understand the various design patterns and services that help improve workload performance. As part of the analysis, identify what you could trade to achieve higher performance. For example, using a cache service can help to reduce the load placed on database systems; however, it requires some engineering to implement safe caching or possible introduction of eventual consistency in some areas.
- **Identify how tradeoffs impact customers and efficiency**: When evaluating performance-related improvements, determine which choices will impact your customers and workload efficiency. For example, if using a key-value data store increases system performance, it is important to evaluate how the eventually consistent nature of it will impact customers.
- **Measure the impact of performance improvements**: As changes are made to improve performance, evaluate the collected metrics and data. Use this information to determine impact that the performance improvement had on the workload, the workload's components, and your customers. This measurement helps you understand the improvements that result from the tradeoff, and helps you determine if any negative side-effects were introduced.
- **Use various performance-related strategies**: Where applicable, utilize multiple strategies to improve performance. For example, using strategies like caching data to prevent excessive network or database calls, using read-replicas for database engines to improve read rates, sharding or compressing data where possible to reduce data volumes, and buffering and streaming of results as they are available to avoid blocking.

Cost Optimization

Topics

Practice Cloud Financial Management

COST 1 How do you implement cloud financial management?

Implementing Cloud Financial Management enables organizations to realize business value and financial success as they optimize their cost and usage and scale on AWS.

Best Practices:

- **Establish a cost optimization function**: Create a team that is responsible for establishing and maintaining cost awareness across your organization. The team requires people from finance, technology, and business roles across the organization.
- **Establish a partnership between finance and technology**: Involve finance and technology teams in cost and usage discussions at all stages of your cloud journey. Teams regularly meet and discuss topics such as organizational goals and targets, current state of cost and usage, and financial and accounting practices.
- **Establish cloud budgets and forecasts**: Adjust existing organizational budgeting and forecasting processes to be compatible with the highly variable nature of cloud costs and usage. Processes must be dynamic using trend based or business driver-based algorithms, or a combination.
- **Implement cost awareness in your organizational processes**: Implement cost awareness into new or existing processes that impact usage, and leverage existing processes for cost awareness. Implement cost awareness into employee training.
- **Report and notify on cost optimization**: Configure AWS Budgets to provide notifications on cost and usage against targets. Have regular meetings to analyze this workload's cost efficiency and to promote cost aware culture.
- **Monitor cost proactively**: Implement tooling and dashboards to monitor cost proactively for the workload. Do not just look at costs and categories when you receive notifications. This helps to identify positive trends and promote them throughout your organization.
- **Keep up to date with new service releases**: Consult regularly with experts or APN Partners to consider which services and features provide lower cost. Review AWS blogs and other information sources.

Expenditure and usage awareness

COST 2 How do you govern usage?

Establish policies and mechanisms to ensure that appropriate costs are incurred while objectives are achieved. By employing a checks-and-balances approach, you can innovate without overspending.

Best Practices:

- **Develop policies based on your organization requirements**: Develop policies that define how resources are managed by your organization. Policies should cover cost aspects of resources and workloads, including creation, modification and decommission over the resource lifetime.
- **Implement goals and targets**: Implement both cost and usage goals for your workload. Goals provide direction to your organization on cost and usage, and targets provide measurable outcomes for your workloads.
- **Implement an account structure**: Implement a structure of accounts that maps to your organization. This assists in allocating and managing costs throughout your organization.
- **Implement groups and roles**: Implement groups and roles that align to your policies and control who can create, modify, or decommission instances and resources in each group. For example, implement development, test, and production groups. This applies to AWS services and third-party solutions.
- **Implement cost controls**: Implement controls based on organization policies and defined groups and roles. These ensure that costs are only incurred as defined by organization requirements: for example, control access to regions or resource types with IAM policies.
- **Track project lifecycle**: Track, measure, and audit the lifecycle of projects, teams, and environments to avoid using and paying for unnecessary resources.

COST 3 How do you monitor usage and cost?

Establish policies and procedures to monitor and appropriately allocate your costs. This allows you to measure and improve the cost efficiency of this workload.

Best Practices:

- **Configure detailed information sources**: Configure the AWS Cost and Usage Report, and Cost Explorer hourly granularity, to provide detailed cost and usage information. Configure your workload to have log entries for every delivered business outcome.
- **Identify cost attribution categories**: Identify organization categories that could be used to allocate cost within your organization.
- **Establish organization metrics**: Establish the organization metrics that are required for this workload. Example metrics of a workload are customer reports produced or web pages served to customers.
- **Configure billing and cost management tools**: Configure AWS Cost Explorer and AWS Budgets inline with your organization policies.
- **Add organization information to cost and usage**: Define a tagging schema based on organization, and workload attributes, and cost allocation categories. Implement tagging across all resources. Use Cost Categories to group costs and usage according to organization attributes.
- **Allocate costs based on workload metrics**: Allocate the workload's costs by metrics or business outcomes to measure workload cost efficiency. Implement a process to analyze the AWS Cost and Usage Report with Amazon Athena, which can provide insight and charge back capability.

COST 4 How do you decommission resources?

Implement change control and resource management from project inception to end-of-life. This ensures you shut down or terminate unused resources to reduce waste.

Best Practices:

- **Track resources over their life time**: Define and implement a method to track resources and their associations with systems over their life time. You can use tagging to identify the workload or function of the resource.
- **Implement a decommissioning process**: Implement a process to identify and decommission orphaned resources.
- **Decommission resources**: Decommission resources triggered by events such as periodic audits, or changes in usage. Decommissioning is typically performed periodically, and is manual or automated.
- **Decommission resources automatically**: Design your workload to gracefully handle resource termination as you identify and decommission non-critical resources, resources that are not required, or resources with low utilization.

Cost-effective resources

COST 5 How do you evaluate cost when you select services?

Amazon EC2, Amazon EBS, and Amazon S3 are building-block AWS services. Managed services, such as Amazon RDS and Amazon DynamoDB, are higher level, or application level, AWS services. By selecting the appropriate building blocks and managed services, you can optimize this workload for cost. For example, using managed services, you can reduce or remove much of your administrative and operational overhead, freeing you to work on applications and business-related activities.

Best Practices:

- **Identify organization requirements for cost**: Work with team members to define the balance between cost optimization and other pillars, such as performance and reliability, for this workload.
- **Analyze all components of this workload**: Ensure every workload component is analyzed, regardless of current size or current costs. Review effort should reflect potential benefit, such as current and projected costs.
- **Perform a thorough analysis of each component**: Look at overall cost to the organization of each component. Look at total cost of ownership by factoring in cost of operations and management, especially when using managed services. Review effort should reflect potential benefit: for example, time spent analyzing is proportional to component cost.
- **Select software with cost effective licensing**: Open source software will eliminate software licensing costs, which can contribute significant costs to workloads. Where licensed software is required, avoid licenses bound to arbitrary attributes such as CPUs, look for licenses that are bound to output or outcomes. The cost of these licenses scales more closely to the benefit they provide.
- **Select components of this workload to optimize cost in line with organization priorities**: Factor in cost when selecting all components. This includes using application level and managed services, such as Amazon RDS, Amazon DynamoDB, Amazon SNS, and Amazon SES to reduce overall organization cost. Use serverless and containers for compute, such as AWS Lambda, Amazon S3 for static websites, and Amazon ECS. Minimize license costs by using open source software, or software that does not have license fees: for example, Amazon Linux for compute workloads or migrate databases to Amazon Aurora.
- **Perform cost analysis for different usage over time**: Workloads can change over time. Some services or features are more cost effective at different usage levels. By performing the analysis on each component over time and at projected usage, you ensure the workload remains cost effective over its lifetime..

COST 6 How do you meet cost targets when you select resource type, size and number?

Ensure that you choose the appropriate resource size and number of resources for the task at hand. You minimize waste by selecting the most cost effective type, size, and number.

Best Practices:

- **Perform cost modeling**: Identify organization requirements and perform cost modeling of the workload and each of its components. Perform benchmark activities for the workload under different predicted loads and compare the costs. The modeling effort should reflect potential benefit: for example, time spent is proportional to component cost.
- **Select resource type and size based on data**: Select resource size or type based on data about the workload and resource characteristics: for example, compute, memory, throughput, or write intensive. This selection is typically made using a previous version of the workload (such as an on-premises version), using documentation, or using other sources of information about the workload.
- **Select resource type and size automatically based on metrics**: Use metrics from the currently running workload to select the right size and type to optimize for cost. Appropriately provision throughput, sizing, and storage for services such as Amazon EC2, Amazon DynamoDB, Amazon EBS (PIOPS), Amazon RDS, Amazon EMR, and networking. This can be done with a feedback loop such as automatic scaling or by custom code in the workload.

COST 7 How do you use pricing models to reduce cost?

Use the pricing model that is most appropriate for your resources to minimize expense.

Best Practices:

- **Perform pricing model analysis**: Analyze each component of the workload. Determine if the component and resources will be running for extended periods (for commitment discounts), or dynamic and short running (for spot or on-demand). Perform an analysis on the workload using the Recommendations feature in AWS Cost Explorer.
- **Implement regions based on cost**: Resource pricing can be different in each region. Factoring in region cost ensures you pay the lowest overall price for this workload
- **Select third party agreements with cost efficient terms**: Cost efficient agreements and terms ensure the cost of these services scales with the benefits they provide. Select agreements and pricing that scale when they provide additional benefits to your organization.
- **Implement pricing models for all components of this workload**: Permanently running resources should utilize reserved capacity such as Savings Plans or reserved Instances. Short term capacity is configured to use Spot Instances, or Spot Fleet. On demand is only used for short-term workloads that cannot be interrupted and do not run long enough for reserved capacity, between 25% to 75% of the period, depending on the resource type.
- **Perform pricing model analysis at the master account level**: Use Cost Explorer Savings Plans and Reserved Instance recommendations to perform regular analysis at the master account level for commitment discounts.

COST 8 How do you plan for data transfer charges?

Ensure that you plan and monitor data transfer charges so that you can make architectural decisions to minimize costs. A small yet effective architectural change can drastically reduce your operational costs over time.

Best Practices:

- **Perform data transfer modeling**: Gather organization requirements and perform data transfer modeling of the workload and each of its components. This identifies the lowest cost point for its current data transfer requirements.
- **Select components to optimize data transfer cost**: All components are selected, and architecture is designed to reduce data transfer costs. This includes using components such as WAN optimization and Multi-AZ configurations
- **Implement services to reduce data transfer costs**: Implement services to reduce data transfer: for example, using a CDN such as Amazon CloudFront to deliver content to end users, caching layers using Amazon ElastiCache, or using AWS Direct Connect instead of VPN for connectivity to AWS.

Manage demand and supply resources

COST 9 How do you manage demand, and supply resources?

For a workload that has balanced spend and performance, ensure that everything you pay for is used and avoid significantly underutilizing instances. A skewed utilization metric in either direction has an adverse impact on your organization, in either operational costs (degraded performance due to over-utilization), or wasted AWS expenditures (due to over-provisioning).

Best Practices:

- **Perform an analysis on the workload demand**: Analyze the demand of the workload over time. Ensure the analysis covers seasonal trends and accurately represents operating conditions over the full workload lifetime. Analysis effort should reflect potential benefit: for example, time spent is proportional to the workload cost.
- **Implement a buffer or throttle to manage demand**: Buffering and throttling modify the demand on your workload, smoothing out any peaks. Implement throttling when your clients perform retries. Implement buffering to store the request and defer processing until a later time. Ensure your throttles and buffers are designed so clients receive a response in the required time.
- **Supply resources dynamically**: Resources are provisioned in a planned manner. This can be demandbased, such as through automatic scaling, or time-based, where demand is predictable and resources are provided based on time. These methods result in the least amount of over or under provisioning.

Optimize over time

COST 10 How do you evaluate new services?

As AWS releases new services and features, it's a best practice to review your existing architectural decisions to ensure they continue to be the most cost effective.

Best Practices:

- **Develop a workload review process**: Develop a process that defines the criteria and process for workload review. The review effort should reflect potential benefit: for example, core workloads or workloads with a value of over 10% of the bill are reviewed quarterly, while workloads below 10% are reviewed annually.

- **Review and analyze this workload regularly**: Existing workloads are regularly reviewed as per defined processes.

AWS Well-Architected Framework

Notices